mathematics

IMPACT FACTOR 2.2

CITESCORE 4.6

Article

Quantum Secure Authentication and Key Exchange Protocol for UAV-Assisted VANETs

Hyewon Park and Yohan Park

Special Issue

Advances in Mathematical Cryptography and Information Security with Applications

Edited by

Prof. Dr. Muzafer Saracevic

MDPI

https://doi.org/10.3390/math14050820

mathematics

MDPI

Article

Quantum Secure Authentication and Key Exchange Protocol for UAV-Assisted VANETs

Hyewon Park and Yohan Park *

School of Computer Engineering, Keimyung University, Daegu 42601, Republic of Korea; wldnjsfuf@stu.kmu.ac.kr
* Correspondence: yhpark@kmu.ac.kr; Tel.: +82-53-580-5229

Abstract

The integration of unmanned aerial vehicles (UAVs) into vehicular ad hoc networks (VANETs) has emerged as a promising solution to overcome the limited coverage of conventional roadside unit (RSU)-based infrastructures. However, UAVs operate in open environments and cannot be fully trusted, while the rapid advancement of quantum computing threatens the long-term security of classical public-key cryptographic systems. As a result, many existing UAV-based VANET authentication schemes face fundamental limitations in future deployments. Most existing schemes either lack post-quantum security or incur excessive computational and communication overhead, making them unsuitable for real-time and high-mobility vehicular environments. In addition, the common assumptions of trusted UAVs do not align with realistic threat models. To address these issues, this paper proposes a lightweight post-quantum authentication and key exchange protocol based on the module learning with errors (MLWE) problem and physically unclonable functions (PUFs). The proposed scheme treats UAVs as untrusted relay nodes and excludes them from session key generation. Its security is evaluated using informal analysis, the real-or-random (RoR) model, BAN logic, and AVISPA, while performance evaluation indicates improved efficiency compared to existing schemes.

Keywords: vehicular ad hoc networks; unmanned aerial vehicles; post-quantum cryptography; module learning with errors; physical unclonable function; mutual authentication; authentication and key exchange

MSC: 94A60; 81P94; 68P25

1. Introduction

The advancement of quantum computing has demonstrated the potential to solve the integer factorization and discrete logarithm problems that underpin public-key cryptography, as shown by Shor in 1994 [1] and Grover in 1996 [2]. Consequently, there has been an increasing need for research on post-quantum cryptography capable of replacing conventional public-key systems [3–5]. Among post-quantum cryptographic schemes, lattice-based cryptography derives its security from the mathematical hardness of high-dimensional lattice problems and offers high computational efficiency since its operations mainly consist of matrix additions and multiplications. Due to its strong resistance to quantum-computer-based attacks as well as its efficiency, lattice-based cryptography is regarded as one of the most promising approaches in modern cryptography [6–9].

Academic Editor: Muzafer Saracevic

Received: 15 January 2026
Revised: 17 February 2026
Accepted: 25 February 2026
Published: 28 February 2026

Research on authentication schemes utilizing lattice-based post-quantum cryptography (PQC) is actively progressing across various fields, and among them, vehicular ad hoc networks (VANETs) are regarded as one of the most critical application domains. VANETs are a core component of smart urban transportation systems that enhance traffic efficiency and safety through communication between vehicles and roadside units (RSUs). If security vulnerabilities exist in this network, the entire urban traffic management infrastructure may face serious disruption [10]. For example, replay attacks and Sybil attacks can inject false traffic information into the system, potentially causing large-scale traffic congestion or interfering with the priority routes of emergency vehicles. In addition, eclipse attacks can isolate a specific vehicle from the network and force it to receive incomplete and adversary-manipulated information instead of legitimate data. Such situations may cause autonomous vehicles to make decisions based on incorrect information, which could ultimately lead to severe accidents. Therefore, VANET security is not merely a matter of traffic control but a fundamental requirement that must be ensured to maintain the overall safety and reliability of the urban transportation ecosystem. However, due to the inherent characteristics of VANETs, where vehicles move at high speeds, it is necessary to achieve a balance between security and efficiency by considering not only security but also the suitability for communication scenarios, fast authentication speed, and continuity of network connectivity [11–13].

Meanwhile, in addition to security vulnerabilities, the conventional VANET architecture also suffers from structural limitations. Since RSUs are installed at fixed locations, vehicles may move outside the communication coverage of RSUs when road structures change or urban areas expand, which can result in communication failure. Furthermore, if an excessive number of vehicles are concentrated on a particular RSU, communication delays may occur due to increased computational load. In addition, if physically installed RSUs are damaged by disasters, all vehicles within the affected area may experience communication outages. Consequently, although authentication is a highly critical component in VANETs, the current architectural constraints raise the possibility that the authentication process itself may not be performed reliably. To address these problems, an architecture that integrates unmanned aerial vehicles (UAVs) into the VANET environment has been proposed. Unlike fixed RSUs, UAVs can be rapidly deployed to appropriate locations when necessary and can dynamically move according to traffic density and road conditions. As a result, issues such as coverage gaps, overload, and communication failures caused by physical damage to conventional RSUs can be effectively mitigated. Figure 1 schematically demonstrates how these limitations in traditional VANETs can be alleviated through the use of UAVs. UAVs can extend communication coverage, distribute the traffic load of congested RSUs, and temporarily replace malfunctioning or damaged RSUs, thereby enhancing the overall reliability and service continuity of the network [14].

However, UAVs are more vulnerable to threats such as eavesdropping, spoofing, node compromise, and man-in-the-middle (MITM) attacks than traditional RSUs, since they operate in open environments, have limited physical protection, and rely on wireless communication links. Therefore, to integrate UAVs into VANETs, it is essential to develop authentication and key exchange (AKE) mechanisms that are both robust and efficient. Nevertheless, many of the currently proposed schemes still depend on traditional public-key cryptography, which cannot withstand future quantum-computer-based attacks, and they also suffer from insufficient authentication structures between vehicles and UAVs, making them vulnerable to impersonation attacks. Since UAV-assisted vehicular communication must support real-time decision making and high mobility, an AKE protocol must ensure computational efficiency while providing security not only against existing network threats but also against future quantum attacks. Accordingly, this paper proposes an AKE

scheme based on module learning with errors (MLWE) that satisfies the characteristics of the VANET environment while providing resilience against future quantum-computer based attacks.

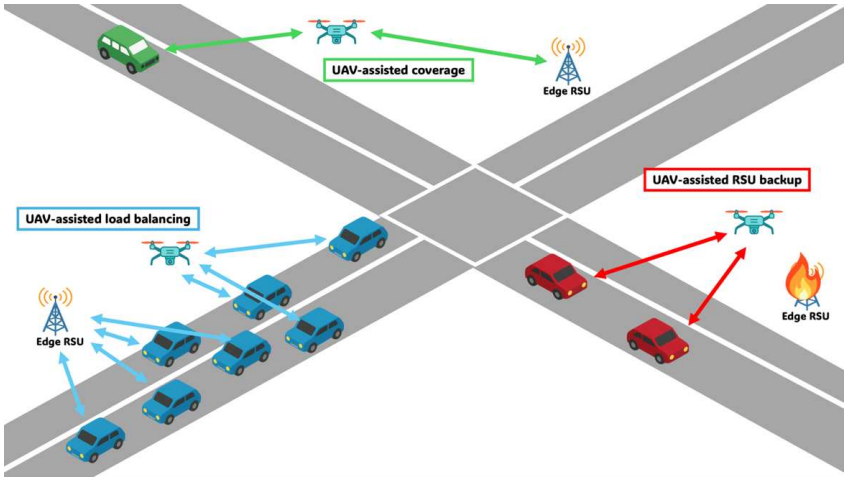


Figure 1. UAV-assisted solutions for issues in VANET environments.

1.1. Contributions

The main contributions of this paper are as follows:

- We clearly analyze the limitations of the conventional RSU-based VANET architecture and justify the introduction of a UAV-assisted structure as an effective solution.
- To ensure secure AKE in the proposed architecture, we design an AKE scheme that combines an MLWE-based key encapsulation mechanism (KEM) and a physical unclonable function (PUF), thereby achieving both security and efficiency.
- We verify the security of the proposed scheme through formal and informal analyses using AVISPA, the real-or-random (RoR) model, and Burrows–Abadi–Needham (BAN) logic.
- Through a comparative evaluation with existing studies, we demonstrate that the proposed scheme outperforms previous approaches in terms of both security and efficiency.

1.2. Organization

The remainder of this paper is organized as follows. In Section 2, we analyze how UAV-assisted VANETs have evolved through prior research and examine their remaining limitations. In Section 3, we introduce MLWE and PUFs, which are utilized for AKE in the proposed scheme, and we summarize the attack models used for security analysis along with the notation employed throughout this paper. In Section 4, we present the proposed system model and describe the AKE procedures in detail within this model. In Section 5, we analyze the security of the proposed scheme using both informal and formal methods. In Section 6, we evaluate the superiority of the proposed scheme in terms of security features and computational cost by comparing it with existing VANET authentication protocols. Finally, in Section 7, we present the conclusion of this paper and discuss future research directions.

2. Related Works

VANETs are a core enabling technology of intelligent transportation systems (ITSs). However, conventional VANET infrastructures mainly rely on fixed RSUs, which results in limited communication coverage, increased deployment and maintenance costs, and performance degradation in environments with many physical obstacles or high disaster

vulnerability. Due to these structural constraints, traditional VANETs face challenges in ensuring large-scale connectivity with high reliability and low latency. As an alternative to overcome these limitations, the integration of UAVs into VANETs, referred to as UAV-assisted VANETs, has been proposed. This approach has emerged as a significant technological advancement that extends communication coverage and enhances network resilience through a mobile aerial infrastructure. Although UAV-assisted VANETs have become an essential component of ITSs due to these advantages, various security limitations have continuously been raised throughout their development.

From 2016 to 2017, research in this area was in its early stage and primarily focused on communication topology and routing management, while security issues were relatively overlooked. Gupta et al. [15] pointed out that UAV networks operate in highly dynamic environments, fundamentally different from traditional MANETs and VANETs, due to rapidly changing topology, intermittent connectivity, and limited energy resources. They argued that such characteristics are difficult to adequately address using existing network architectures alone. However, their study did not propose specific authentication or cryptographic mechanisms. Similarly, Menouar et al. [16] proposed an architecture that utilizes UAVs as an extended infrastructure to support ITS functionalities such as traffic monitoring, law enforcement, and emergency assistance. Nevertheless, security and privacy protection were not reflected as core design considerations and were instead mentioned only as future research directions. As a result, early UAV-VANET studies largely concentrated on expanding connectivity and improving communication efficiency, while failing to sufficiently address threats such as message falsification, location tracking, and unauthorized access.

Since 2018, UAVs have evolved to serve as auxiliary infrastructure supporting the internet of vehicles (IoV), playing a crucial role in various applications such as data collection, communication relaying, and emergency communication. This transition further emphasized the need for secure communication mechanisms; however, concrete and systematic security designs were still insufficient. For example, Ng et al. [17] discussed the use of UAVs as relay nodes to support federated learning in IoV environments, aiming to enhance data reliability and model integrity. Nonetheless, their work did not sufficiently address potential threats such as adversarial data manipulation or eavesdropping.

As UAV-assisted communication technologies continued to expand, security requirements became increasingly critical, and since 2023, cryptography-based approaches have begun to emerge as a major research trend. El-Zawawy et al. [18] proposed an authentication protocol that integrates ECC with blockchain in an attempt to defend against UAV hijacking and data injection attacks. However, the delay and computational overhead introduced by blockchain verification posed limitations to its applicability in real-time vehicular environments. Similarly, the ECC-based mutual authentication scheme proposed by Miao et al. [19] was criticized for its high computational complexity. In addition, the lightweight authentication method introduced by Cui et al. [20], which utilizes chaotic maps and honeywords, still suffered from synchronization failures and reauthentication overhead in highly mobile environments. Consequently, the trade-off between enhanced security strength and real-time performance and energy efficiency became increasingly evident.

Since 2024, research has expanded beyond purely cryptography-centric approaches toward a broader security paradigm that incorporates trust management and privacy protection. Guo et al. [21] designed a zero-trust-based framework that enables continuous trust verification; however, it did not fully resolve issues related to communication overhead caused by frequent reauthentication and concerns regarding personal data collection. More recently, Choi et al. [22] and Lin et al. [23] proposed lightweight security and privacy-preserving mechanisms that combine PUFs with the concept of dynamic identities. Nevertheless, challenges such as synchronization inconsistencies in multi-UAV environ-

ments, insider threats, and the lack of standardized key management remain unresolved. In addition, Telikani et al. [24] highlighted that despite the rapid progress of UAV–ITS integration, the absence of unified security standards and interoperable authentication frameworks continues to be a major obstacle to large-scale practical deployment.

In summary, research on UAV-assisted VANETs has gradually evolved from its initial focus on extending connectivity and designing network architectures to a security-oriented direction that encompasses cryptography-based authentication schemes, trust management, and privacy-preserving mechanisms. However, a lightweight and highly reliable security framework that simultaneously satisfies the characteristics of UAV environments, including real-time constraints, high mobility, and resource limitations, has not yet been fully established. Moreover, the lack of unified security standards and interoperability issues remains a significant barrier to practical deployment. In addition, most existing studies rely on classical public-key cryptographic systems such as ECC, and therefore fail to address the fundamental vulnerabilities that arise in quantum computing environments. Considering these factors, it is essential to develop a new security framework for UAV-assisted VANETs that ensures lightweight operation, real-time performance, and scalability, while also maintaining a stable level of security in the post-quantum era. To meet this need, this study focuses on the design of quantum-resistant security mechanisms for UAV-assisted VANET environments and aims to address the limitations of existing security research.

Table 1 summarizes the previously discussed studies on UAV-assisted VANETs, providing an overview of the technological development trends in this field and the limitations of existing security research.

Table 1. Overview of research trends and security limitations in UAV-assisted VANETs.

Author	Main Contributions	Limitations/Challenges
Gupta et al. [15]	Identified UAV networks as fundamentally different from MANETs/VANETs due to high mobility, intermittent connectivity, and limited energy.	Did not propose authentication or encryption mechanisms; focused mainly on topology characteristics.
Menouar et al. [16]	Proposed UAVs as extended ITS infrastructure supporting traffic monitoring, law enforcement, and emergency assistance.	Security and privacy not incorporated as core design factors; only mentioned as future research directions.
Ng et al. [17]	Utilized UAVs as relay nodes in IoV to support federated learning and enhance data reliability and model integrity.	Did not sufficiently address adversarial data manipulation or eavesdropping threats.
El-Zawawy et al. [18]	Developed BDIVE protocol combining ECC and blockchain to defend against UAV hijacking and data injection.	Blockchain verification introduces high latency and computational overhead, limiting real-time applicability.
Miao et al. [19]	Designed ECC-based mutual authentication protocol for UAV–vehicle secure communication.	Computational complexity too high for resource-constrained UAV environments.
Cui et al. [20]	Proposed lightweight authentication using chaotic maps and honeywords for UAV–vehicle communication.	Synchronization failures and reauthentication overhead in highly dynamic mobility environments.
Guo et al. [21]	Introduced zero-trust-based UAVA framework for continuous trust verification.	Frequent reauthentication causes communication overhead; privacy concerns due to continuous data collection.
Choi et al. [22]	Proposed lightweight authentication using PUF and dynamic identity for UAV environments.	Synchronization mismatches, insider threats, and multi-UAV trust management challenges remain.
Lin et al. [23]	Proposed DIHE protocol enabling mutual authentication and anonymity between UAVs and control stations.	Lack of standardized key management limits interoperability and deployment scalability.
Telikani et al. [24]	Analyzed UAV–ITS integration status and categorized UAV roles across multiple dimensions.	Pointed out the absence of unified security standards and interoperable frameworks as key deployment barriers.

3. Preliminaries

3.1. MLWE

Lattice-based cryptography originates from the learning with errors (LWE) problem proposed by Regev [25], whose average-case hardness is reducible to worst-case lattice problems such as SVP and SIVP. However, conventional LWE-based schemes suffer from inefficiencies due to high-dimensional matrix operations, which motivated the introduction of the ring learning with errors (RLWE) problem by Peikert [26] to improve efficiency using polynomial ring structures. Despite its efficiency benefits, the ring structure in RLWE introduces additional algebraic constraints that may raise concerns regarding structural attacks and limit flexibility in cryptographic design. MLWE further generalizes both LWE and RLWE, providing a flexible framework that balances security and efficiency by combining the matrix-based structure of LWE with the ring structure of RLWE. Langlois and Stehlé [27] showed that MLWE admits average-case to worst-case reductions over standard lattice problems and naturally reduces to RLWE when $k = 1$ and to LWE when $R_q = \mathbb{Z}_q$.

3.1.1. Definition of the MLWE Problem

The MLWE problem is parameterized by (n, q, k, χ) over the polynomial ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$. Here, n denotes the degree of the polynomial ring, q is a prime modulus that defines the finite field over which the polynomial coefficients are represented, k denotes the dimension of the polynomial vector, and χ specifies the probability distribution from which the error terms are sampled. Let $A \in R_q^{k \times k}$ be sampled uniformly at random, and let secret vectors s and error vectors e be sampled from the distribution χ. The MLWE assumption states that distinguishing valid MLWE samples $(A, As + e)$ from uniformly random samples is computationally infeasible. The hardness of MLWE relies on worst-case lattice problems, providing strong security guarantees in the post-quantum setting.

3.1.2. Crystals-Kyber KEM

In this paper, in order to design the proposed protocol based on the security of MLWE, we adopt the MLWE-based KEM construction proposed in Crystals-Kyber [28]. Crystals-Kyber was ultimately selected as the standard algorithm for the KEM mechanism in the NIST PQC standardization process, and its security and efficiency have been extensively evaluated.

A KEM is a technique that enables two parties to securely establish a shared secret using an asymmetric cryptographic system. According to the official specification, Crystals-Kyber achieves IND-CCA2 (indistinguishability under adaptive chosen-ciphertext attack) security by applying the Fujisaki–Okamoto (FO) transformation to a public-key encryption scheme that is IND-CPA (indistinguishable under chosen-plaintext attack) secure.

Crystals-Kyber consists of the following procedures: key generation, encapsulation, and decapsulation.

- Key generation: Secret vectors $s, e \leftarrow \chi^k$ are sampled, and for a random matrix $A \in R_q^{k \times k}$, the following values are computed:

$$t = A \cdot s + e, \quad pk = (A, t), \quad sk = s$$

- Encapsulation: Using the receiver's public key $pk = (A, t)$, the sender performs the following procedure. A random message $m \in \{0, 1\}^l$ is chosen, and $r, e_1, e_2 \leftarrow \chi^k$ are sampled. Then, the following values are computed:

$$u = A^\top \cdot r + e_1, \quad v = t^\top \cdot r + e_2 + \text{Encode}(m)$$

Here, u acts as a noise-based random component, while v is responsible for hiding the message. These two values together form the ciphertext, transmitted as

$$ct = (u, v), \quad SK = H(m)$$

- Decapsulation: The receiver recovers the message using the secret key s by computing

$$m' = \text{Decode}(v - u^{\top} \cdot s), \quad SK = H(m')$$

3.1.3. Message Encoding and Decoding

In Crystals-Kyber schemes, the message m is not encrypted directly. Instead, it is embedded into the $\mathbb{Z}_q$ domain and included within the ciphertext. The following encoding and decoding functions are used for this purpose. These transformations improve the robustness of message recovery in the presence of noise and enable secure and efficient session key agreement without requiring a reconciliation step.

- Encode: Each bit m_i of the binary message $m \in \{0,1\}^{\ell}$ is mapped as follows:

$$\text{Encode}(m_i) = \begin{cases} 0 & \text{if } m_i = 0 \\ \left\lfloor \frac{q}{2} \right\rfloor & \text{if } m_i = 1 \end{cases}$$

- Decode: For each polynomial coefficient $x \in \mathbb{Z}_q$ obtained during decryption, the following threshold function is applied to recover the corresponding binary value:

$$\text{Decode}(x) = \begin{cases} 1 & \text{if } x \geq \left\lfloor \frac{q}{4} \right\rfloor \\ 0 & \text{otherwise} \end{cases}$$

3.2. Physical Unclonable Function

A PUF, which is designed as a hardware-based one-way hash function, is a physical microstructure characterized by low computational requirements. A PUF generates unique input–output pairs, referred to as challenge–response pairs, that are inherently formed during the fabrication process of semiconductors and integrated circuits. Recently, PUFs have been widely adopted as a key enabling technology to strengthen authentication and key management mechanisms in various lightweight security environments, including IoT, VANETs, drone networks, and wireless medical sensor networks. PUFs generate responses to given challenges by exploiting inherent, device-specific physical variations, typically using a random bit string as the challenge input. Because the fabrication process introduces uncontrollable randomness—even from the manufacturer's perspective—exact replication is extremely difficult [29–31]. Consequently, each PUF produces a distinctive response to the same challenge, and PUFs are widely regarded as resistant to cloning, thereby helping to mitigate attacks such as device replication and physical capture [32].

However, when PUF stability is limited, environmental conditions (e.g., temperature or voltage fluctuations) can affect response reproducibility. To address this issue, several classes of so-called ideal PUFs have been proposed to improve stability across wide operating ranges. For instance, certain designs exploit the randomness of soft gate-oxide breakdown locations or deliberately introduced permanent physical defects to yield more stable responses [33,34]. By adopting such ideal PUF designs, it is possible to reduce computational and storage overhead and to lessen dependence on stability-enhancement mechanisms, such as helper data or fuzzy extractors [35], while still maintaining practical deployability [36].

In this paper, the challenge value is denoted as CH and the corresponding response value as RE, which is expressed as $RE = PUF(CH)$. The fundamental properties of PUFs are as follows.

- A PUF is a hardware circuit that cannot be cloned, and there exists no $PUF'(CH)$ such that $PUF'(CH) = PUF(CH)$.
- Although $PUF(CH) = RE$ can be easily computed, predicting RE for a given CH within polynomial time is computationally infeasible.
- The output of a PUF is statistically unpredictable. By exploiting the inherent physical randomness of PUFs, intrusive physical attacks can be effectively mitigated.

3.3. Adversary Model

To evaluate the security of the proposed scheme, we adopt the Dolev–Yao (DY) model [37], which is a standard adversarial model in network security analysis. The DY model assumes that an adversary has complete control over the public communication channel, making it well suited for analyzing open and dynamic wireless VANET environments. The specific capabilities and assumptions of the adversary are described as follows.

- The adversary can eavesdrop on, intercept, modify, delete, and replay all messages transmitted over the wireless communication channel. In addition, the adversary can disrupt ongoing sessions or actively initiate new sessions.
- The adversary may register as a legitimate user within the network and communicate lawfully with other nodes. Furthermore, based on collected information, the adversary may impersonate a vehicle, UAV, or RSU to participate in the authentication procedure or generate forged messages.
- The adversary may physically capture a vehicle's on-board unit (OBU) or a UAV and attempt to extract stored internal information. Therefore, resistance against physical capture attacks is also considered.
- Although the UAV performs message-relaying and verification functions, it is not assumed to be fully trusted. That is, the UAV is considered a potential attack vector that may behave maliciously or leak stored data. However, by design, the UAV is strictly prevented from directly accessing core secret information, such as users' real identities or session keys.
- The adversary may obtain encrypted messages; however, under the assumption of a secure public-key cryptosystem, the adversary cannot recover the plaintext without possessing the corresponding decryption key.
- The adversary may perform offline guessing attacks, such as repeated hash computations, based on captured messages. Nevertheless, the protocol is designed such that user identity information and secret values are not simultaneously exposed, and disclosure of a single piece of information does not enable the recovery of meaningful secret data.

It is further assumed that the trusted authority is not compromised and that the underlying cryptographic primitives, such as the employed public-key encryption and hash functions, are secure. Attacks targeting physical-layer communication, including jamming or denial-of-service (DoS), as well as side-channel attacks are considered beyond the scope of this work.

3.4. Notation

Table 2 summarizes the notation and symbols used throughout the proposed protocol.

Table 2. Notation used in the scheme.

Notation	Description
V_i, U_j, R_l	i-th vehicle, j-th UAV, l-th RSU
VID_i, UID_j, RID_l	Identifier of V_i, U_j, R_l
TID_i	Pseudonym identifier of V_i
A^d	Public matrix $A \in \mathbb{R}^{k \times l}$
s, s^d	Secret key of RA, R^d
p, p^d	Public key of RA, R^d
$c_i = (u_i, v_i)$	Ciphertext
k_i	Shared secret key
Encode$(\cdot)$	Encoding function
Decode$(\cdot)$	Decoding function
CH_i, CH_j, CH_l^r	PUF challenge of V_i, U_j, R_l
RE_i, RE_j, RE_l^r	PUF response of V_i, U_j, R_l
TS_1, TS_2, TS_3, TS_4	Timestamps
SK_a	Session key
$h(\cdot)$	Hash function
$\oplus$	Bitwise XOR operation
$\|\|$	Concatenation operator
u^t	Transpose of vector u

4. Proposed Scheme

We aim to overcome the limitations of conventional public-key cryptographic systems under quantum computing threats and to establish a secure and efficient AKE mechanism for UAV-assisted VANETs. To achieve this objective, we adopt an MLWE-based KEM to ensure resistance against quantum attacks, while maintaining lightweight operation and real-time communication performance suitable for the dynamic characteristics of VANETs. In this section, we first present the overall architecture of the proposed system and the roles of the participating entities. We then describe the initialization and registration procedures, followed by the detailed AKE execution process.

4.1. Communication Entities

The descriptions of the entities participating in the communication are as follows.

4.1.1. Trusted Authority

The trusted authority (TA) is responsible for system initialization and the registration procedures of RSUs, UAVs, and vehicles. The TA distributes the essential parameters that each entity must possess for AKE. Any entity that intends to participate in communication must complete the registration process through the TA before engaging in legitimate communication.

4.1.2. Roadside Units

RSUs are essential infrastructure installed along roadways to enable real-time communication with vehicles. However, they are vulnerable to physical attacks, may be damaged in disaster or emergency situations, and their fixed deployment makes it difficult to adjust coverage. To address these limitations, UAVs can be utilized as auxiliary communication support units. When vehicles receive messages relayed by UAVs, rigorous authentication is required to ensure the reliability of the transmitted information, since inadequate authentication may allow adversaries to introduce malicious UAVs that are difficult to distinguish from legitimate ones. Meanwhile, even when UAVs assist communication, the responsibil-

ity for processing vehicular information still remains with the RSUs, and therefore RSUs continually perform the role of establishing session keys with vehicles.

4.1.3. Unmanned Aerial Vehicles

UAVs are employed as auxiliary devices in situations where RSUs are unable to maintain communication or require additional support, and they function as relays that forward messages received from vehicles to the RSUs. However, similar to RSUs, UAVs are vulnerable to physical attacks. Therefore, sensitive or confidential information is not stored on UAVs, and since the RSUs are the final recipients of the messages, UAVs cannot access or interpret their contents. Consequently, UAVs only assist in the authentication process between vehicles and RSUs and do not participate in session key generation.

4.1.4. Vehicles

All vehicles are equipped with an OBU and can participate in communication after completing the user login process using the information obtained during registration. As the entity responsible for establishing the session key with the RSU, the vehicle initiates the entire session. Since UAVs are regarded as untrusted entities, the real identity of the vehicle is protected during the authentication process, and only the RSU is able to verify it. Ultimately, the session key is exclusively established between the vehicle and the RSU, ensuring that the UAV is completely excluded from the key exchange process.

Figure 2 illustrates the overall system architecture. It shows how the TA, RSUs, UAVs, and vehicles interact with each other through the initialization, registration, and authentication processes.

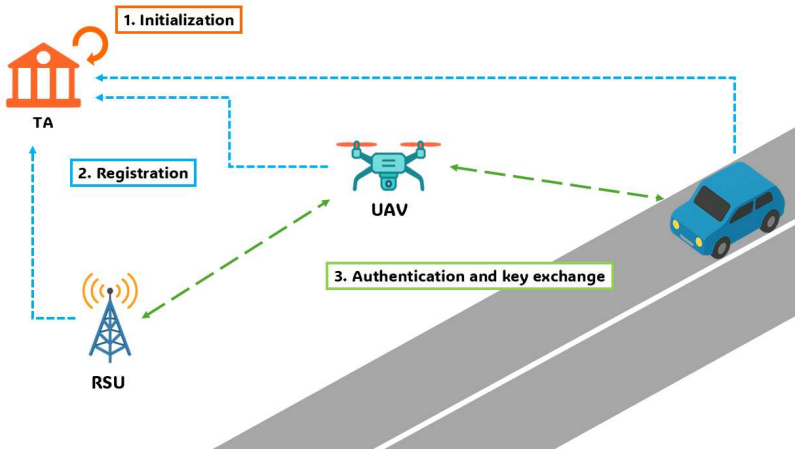


Figure 2. System architecture.

4.2. Initialization Phase

Before the entire network begins operation, the TA performs an initialization process to configure the public parameters required for secure communication. Figure 3 shows the initialization process of the TA, and the detailed procedure is as follows. The TA first selects a public matrix $A \in R_q^{k \times k}$ and a cryptographic hash function $h : 0,1^* \rightarrow 0,1^k$, which maps an input of arbitrary length to a fixed-length output. Then, the TA samples the secret key and error vector $s, e \leftarrow \chi^k$ and computes the corresponding public key as $P = A \cdot s + e$. After completing these steps, the TA publishes the system parameters $A, h(\cdot), P, \chi^k$, so that they can be used by all network entities.

TA
$A \leftarrow R_q^{k \times k}$
$h : \{0,1\}^* \rightarrow \{0,1\}^k$
$s, e \leftarrow \chi^k$
$P = A \cdot s + e$
Publish $\left\{A, h(\cdot), P, \chi^k\right\}$

Figure 3. TA initialization phase.

4.3. Registration Phase

All RSUs, UAVs, and vehicles that intend to participate in communication are required to complete a registration process with the TA.

4.3.1. RSU Registration

The registration process of the RSU is illustrated in Figure 4, and the detailed procedure is as follows.

Step 1: The RSU R_l selects a unique identifier RID_l and randomly samples two vectors s_l and e_l from the distribution χ^k. It then computes $P_l = A \cdot s_l + e_l$. The pair $\{RID_l, P_l\}$ is transmitted to the TA through a secure channel.

Step 2: Upon receiving RID_l from R_l, the TA generates a challenge value CH_l and sends it to R_l via a secure channel.

Step 3: After receiving the challenge value CH_l, R_l generates the response $RE_l = PUF(CH_l)$ using its internal PUF. To conceal the secret key, R_l computes $S_l = s_l \oplus RE_l$. Finally, R_l stores the set $\{RID_l, CH_l, P_l, S_l\}$ for future authentication and verification.

R_l	**Secure channel**	TA
Select RID_l $s_l, e_l \leftarrow \chi^k$ $P_l = A \cdot s_l + e_l$		
	$\{RID_l, P_l\} \longrightarrow$	
		Generate CH_l
	$\leftarrow \{CH_l\}$	
$RE_l = PUF(CH_l)$ $S_l = s_l \oplus RE_l$ Store $\{RID_l, CH_l, R_l, S_l\}$		

Figure 4. RSU registration phase.

4.3.2. UAV Registration

The registration process of the UAV is illustrated in Figure 5, and the detailed procedure is as follows.

Step 1: The UAV U_j selects a unique identifier UID_j and generates a challenge value CH_j. By applying CH_j to its internal PUF, U_j generates the response $RE_j = PUF(CH_j)$. Using this response, U_j computes $Z_j^1 = h(UID_j \mid RE_j)$. The computed pair $\{UID_j, Z_j^1\}$ is then transmitted to the TA through a secure channel.

Step 2: Upon receiving $\{UID_j, Z_j^1\}$ from U_j, the TA generates a random nonce n_j and forwards the set $\{UID_j, Z_j^1, n_j\}$ to R_l. After receiving it, R_l computes the PUF response $RE_l = PUF(CH_l)$ and then calculates $Z_j^2 = h(UID_j \,||\, n_j \,||\, RE_l \,||\, s_l)$. In addition, R_l generates $Z_j^3 = (n_j \,||\, Z_j^1) \oplus h(RE_l \,||\, s_l)$ and stores the tuple $\{UID_j, Z_j^3\}$ for subsequent authentication.

Step 3: The TA sends Z_j^2 received from R_l to U_j. The UAV then computes $Z_j^4 = Z_j^1 \oplus Z_j^2$ and stores the set $\{UID_j, CH_j, Z_j^4\}$ to complete the registration procedure.

U_j	**Secure channel**	TA
Select U_j		
Generate CH_j		
Compute		
$RE_j = PUF(CH_j)$		
$Z_j^1 = h(UID_j \| RE_j)$		
	$\{UID_j, Z_j^1\} \longrightarrow$	
		Generate n_j
		Send $\{UID_j, Z_j^1, n_j\}$ to R_l
		RSU computes
		$RE_l = PUF(CH_l)$
		$Z_j^2 = h(UID_j \| n_j \| RE_l \| s_l)$
		$Z_j^3 = (n_j \| Z_j^1) \oplus h(RE_l \| s_l)$
		RSU store UID_j, Z_j^3
	$\{Z_j^2\} \longleftarrow$	
Compute $Z_j^4 = Z_j^1 \oplus Z_j^2$		
Store $\{UID_j, CH_j, Z_j^4\}$		

Figure 5. UAV registration phase.

4.3.3. Vehicle Registration

The vehicle registration process is shown in Figure 6, and the detailed procedure is as follows.

V_i	**Secure channel**	TA
Select VID_i		
	$\{VID_i\} \longrightarrow$	
		Generate CH_i, n_i
		Send $\{VID_i, n_i\}$ to R_l
		RSU computes
		$RE_l = PUF(CH_l)$
		$Z_i^1 = n_i \oplus h(VID_i \| RE_l)$
		RSU store $\{VID_i, Z_i^1\}$
	$\{n_i\} \longleftarrow$	
Generate s_i		
Compute $RE_i = PUF(CH_i)$		
$Z_i^2 = n_i \oplus h(RE_i \| s_i)$		
$S_i = s_i \oplus RE_i$		
Store $\{VID_i, CH_i, Z_i^2, S_i\}$		

Figure 6. Vehicle registration phase.

Step 1: The vehicle V_i selects a unique identifier VID_i and sends it to the TA through a secure channel.

Step 2: Upon receiving VID_i, the TA generates a random nonce n_i and forwards the set $\{VID_i, n_i\}$ to the RSU R_l. Then, R_l applies the challenge CH_l to its PUF to obtain the response $RE_l = PUF(CH_l)$. Using this response, R_l computes $Z_i^1 = n_i \oplus h(VID_i \mid RE_l)$ and stores the pair $\{VID_i, Z_i^1\}$ for subsequent authentication.

Step 3: The TA sends the received n_i to V_i. The vehicle then generates a random value s_i and a challenge CH_i, and computes the corresponding PUF response $RE_i = PUF(CH_i)$. It then calculates $Z_i^2 = n_i \oplus h(RE_i \,||\, s_i)$ and derives $S_i = s_i \oplus RE_i$. Finally, V_i stores the set $\{VID_i, CH_i, Z_i^2, S_i\}$ to complete the registration procedure.

4.4. Authentication and Key Exchange Phase

The vehicle must establish a session key with the RSU while receiving relay support from the UAV for data transmission, and the protocol is designed to enable mutual authentication among the RSU, UAV, and vehicle to ensure secure communication. The AKE procedure is illustrated in Figure 7, and the detailed process is described as follows.

Step 1: The vehicle V_i computes the PUF response $RE_i = PUF(CH_i)$ using its identifier VID_i as input, and derives $S_i = s_i \oplus RE_i$ and $n_i^* = Z_i^2 \oplus h(RE_i \mid s_i)$. It then generates random vectors $r_i, t_i^1, t_i^2 \in \chi^k$ and a message $m_i \in \{0,1\}^k$. After generating a timestamp TS_1, the following computations are performed: $u_i = A^T \cdot r_i + t_i^1$, $v_i = P_i \cdot r_i + t_i^2 + Encode(m_i)$, $c_i = (u_i, v_i)$, and $k_i = h(m_i)$. The vehicle then computes $TID_i = VID_i \oplus h(k_i \,||\, TS_1)$, $V_1 = h(VID_i \,||\, n_i \,||\, n_i^* \,||\, k_i \,||\, TS_1)$, $V_2 = h(TID_i \,||\, UID_j \,||\, n_i)$, and $N_1 = n_i \oplus h(VID_i \,||\, n_i \,||\, k_i \,||\, TS_1)$. Finally, V_i sends $\{TID_i, c_i, V_1, V_2, N_1, TS_1\}$ to U_j.

Step 2: Upon receiving the message, U_j first verifies the validity of TS_1 and then generates its own timestamp TS_2. It subsequently computes $RE_j = PUF(CH_j)$, $Z_j^{1*} = h(UID_j \,||\, RE_j)$, and $Z_j^{2*} = Z_j^3 \oplus Z_j^{1*}$. Next, it computes $V_3 = h(TID_i \,||\, UID_j \,||\, V_1 \,||\, Z_j^{2*} \,||\, TS_2)$. Finally, U_j forwards the set $\{TID_i, UID_j, c_i, V_3, N_1, TS_1, TS_2\}$ to the RSU R_l.

Step 3: Upon receiving the request, R_l verifies the timestamp TS_2 and computes its PUF response $RE_l = PUF(CH_l)$. It then derives $S_i^* = S_l \oplus RE_l$, $n_i = Z_j^3 \oplus h(RE_l \,||\, s_l)$, and $Z_j = h(UID_j \,||\, n_i \,||\, RE_l \,||\, s_l)$. After that, R_l computes $V_3^* = h(TID_i \,||\, UID_j \,||\, V_1 \,||\, Z_j \,||\, TS_2)$ and checks whether $V_3^* = V_3$. If the verification is successful, R_l generates the session key $SK = h(VID_i \,||\, UID_j \,||\, RID_l \,||\, n_i \,||\, k_i \,||\, TS_3)$, computes $N_2 = n_i^* \oplus h(Z_j \,||\, TS_3)$, and generates $V_4 = h(RID_l \,||\, UID_j \,||\, Z_j \,||\, n_i^* \,||\, TS_3)$ and $V_5 = h(RID_l \,||\, VID_i \,||\, SK \,||\, TS_3)$. Finally, R_l sends $\{RID_l, N_2, V_4, V_5, TS_3\}$ to U_j.

Step 4: After receiving the message, U_j first verifies TS_3, and then computes $n_i = N_2 \oplus h(Z_j \,||\, TS_3)$ and $V_4^* = h(RID_l \,||\, UID_j \,||\, Z_j \,||\, n_i \,||\, TS_3)$ to verify whether $V_4^* = V_4$. It also computes $V_2^* = h(TID_i \,||\, UID_j \,||\, n_i)$ and checks whether $V_2^* = V_2$. If all verifications are successful, U_j generates a new timestamp TS_4 and computes $V_6 = h(TID_i \,||\, UID_j \,||\, n_i \,||\, V_5 \,||\, TS_4)$. It then sends $\{RID_l, UID_j, V_5, V_6, TS_4\}$ to V_i.

Step 5: Finally, V_i verifies the freshness of TS_4 and computes $V_6^* = h(TID_i \,||\, UID_j \,||\, n_i \,||\, V_5 \,||\, TS_4)$ to check whether $V_6^* = V_6$. If the verification succeeds, the vehicle derives the session key $SK^* = h(VID_i \,||\, UID_j \,||\, RID_l \,||\, n_i \,||\, k_i \,||\, TS_3)$ and computes $V_5^* = h(RID_l \,||\, VID_i \,||\, SK^* \,||\, TS_3)$ to verify whether $V_5^* = V_5$. Once all verification steps are successfully completed, mutual authentication among V_i, U_j, and R_l is achieved. Thereafter, a secure session key is successfully established between V_i and R_l, ensuring confidential communication.

V_i $(VID_i, CH_i, Z_i^2, S_i)$	U_j (UID_j, CH_j, Z_j^4)	R_l $(RID_l, CH_l, R, S_l, UID_j, Z_j^3, VID_i, Z_i^1)$
Input VID_i		
Calculate		
$RE_i^* = PUF(CH_i)$		
$s_i^* = S_i \oplus RE_i^*$		
$n_i^* = Z_i^2 \oplus h(RE_i^* \| S_i^*)$		
Generate $r_i, t_i^1, t_i^2 \leftarrow \chi^k$, $m_i \leftarrow \{0,1\}^k$		
Random number n_1		
Timestamp TS_1		
Calculate		
$u_i = A^T \cdot r_i + t_i^1$		
$v_i = P_l^T \cdot r_i + t_i^2 + Encode(m_i)$		
$c_i = (u_i, v_i)$		
$k_i = h(m_i)$		
$TID_i = VID_i \oplus h(k_i \| TS_1)$		
$V_1 = h(VID_i \| n_i \| n_1 \| k_i \| TS_1)$		
$V_2 = h(TID_i \| UID_j \| n_1)$		
$N_1 = n_1 \oplus h(VID_i \| n_i \| k_i \| TS_1)$		
$\xrightarrow{\{TID_i, c_i, V_1, V_2, N_1, TS_1\}}$		
	Verify TS_1	
	Timestamp TS_2	
	Save V_2	
	Calculate	
	$RE_j^* = PUF(CH_j)$	
	$Z_j^{1*} = h(UID_j \| RE_j^*)$	
	$Z_j^{2*} = Z_j^4 \oplus Z_j^{1*}$	
	$V_3 = h(TID_i \| UID_j \| V_1 \| Z_j^{2*} \| TS_2)$	
	$\xrightarrow{\{TID_i, UID_j, c_i, V_1, V_3, N_1, TS_1, TS_2\}}$	
		Verify TS_2
		Calculate
		$RE_l^* = PUF(CH_l)$
		$s_l^* = S_l \oplus RE_l^*$
		$(n_j^* \| Z_j^{1**}) = Z_j^3 \oplus h(RE_l^* \| s_l^*)$
		$Z_j^{2**} = h(UID_j \| n_j^* \| RE_l^* \| s_l^*)$
		$V_3^* = h(TID_i \| UID_j \| V_1 \| Z_j^{2**} \| TS_2)$
		Verify $V_3 = V_3^*$
		Calculate
		$m_i^* = Decode(v_i - u_i^T \cdot s_l^*)$
		$k_i^* = h(m_i^*)$
		$VID_i^* = TID_i \oplus h(k_i^* \| TS_1)$
		$n_i^* = Z_i^1 \oplus h(VID_i^* \| RE_l^*)$
		$n_1^* = N_1 \oplus h(VID_i^* \| n_i^* \| k_i^* \| TS_1)$
		$V_1^* = h(VID_i^* \| n_i^* \| n_1^* \| k_i^* \| TS_1)$
		Verify $V_1 = V_1^*$
		Timestamp TS_3
		Calculate
		$SK = h(VID_i^* \| UID_j \| RID_l \| n_1^* \| k_i^* \| TS_3)$
		$N_2 = n_1^* \oplus h(Z_j^{1**} \| TS_3)$
		$V_4 = h(RID_l \| UID_j \| Z_j^{1**} \| n_1^* \| TS_3)$
		$V_5 = h(RID_l \| VID_i^* \| SK \| TS_3)$
	$\xleftarrow{\{RID_l, N_2, V_4, V_5, TS_3\}}$	
	Verify TS_3	
	Calculate	
	$n_1^{**} = N_2 \oplus h(Z_j^{1*} \| TS_3)$	
	$V_4^* = h(RID_l \| UID_j \| Z_j^{1*} \| n_1^{**} \| TS_3)$	
	Verify $V_4 = V_4^*$	
	Calculate	
	$V_2^* = h(TID_i \| UID_j \| n_1^{**})$	
	Verify $V_2 = V_2^*$	
	Timestamp TS_4	
	$V_6 = h(TID_i \| UID_j \| n_1^{**} \| V_5 \| TS_4)$	
$\xleftarrow{\{RID_l, UID_j, V_5, V_6, TS_3, TS_4\}}$		
Verify TS_4		
Calculate		
$V_6^* = h(TID_i \| UID_j \| n_1 \| V_5 \| TS_4)$		
Verify $V_6 = V_6^*$		
Calculate		
$SK^* = h(VID_i \| UID_j \| RID_l \| n_1 \| k_i \| TS_3)$		
$V_5^* = h(RID_l \| VID_i \| SK^* \| TS_3)$		
Verify $V_5 = V_5^*$		

Figure 7. Authentication and key exchange phase.

4.5. Vehicle Revocation

The proposed scheme provides robust resistance against various network and physical attacks by utilizing Crystals-Kyber KEM and PUF. However, practical deployment requires consideration of additional threat scenarios, such as physical OBU damage, UAV capture, PUF spoofing, and long-term secret leakage. To address these issues, we define a dynamic revocation mechanism.

In this framework, a vehicle V_i is designated for revocation if it meets any of the following criteria: (1) extraction of internal stored values due to OBU theft or damage; (2) compromise of long-term secret information; (3) repeated inconsistencies in PUF responses to the same challenge indicating hardware unreliability; or (4) repeated transmission of false information and violations of network operational policies.

Vehicles identified by these conditions are registered in a revocation list (RL) managed by the TA, with the specific procedure for revocation and authentication filtering detailed in Algorithm 1. During the authentication and key establishment process, the RSU verifies the vehicle's identity. If V_i is found to be compromised, the RSU immediately sends a revocation request containing VID_i to the TA. Upon receipt, the TA updates the RL and distributes it to all $RSUs$. Subsequently, any RSU performing authentication cross-references the RL; if VID_i is present, the process is terminated, and the request is rejected to ensure network integrity.

Algorithm 1 Vehicle Authentication and Dynamic Revocation Procedure

Require: V_i, RSU, TA, RL
Ensure: Session Key SK or VID_i added to RL
Step 1: Vehicle Authentication Request
V_i sends authentication request with VID_i to RSU.
Step 2: RL Verification (by RSU)
if $VID_i \in RL$ **then**
  RSU rejects the request and terminates the session. {Vehicle is revoked}
else
  RSU proceeds with authentication.
  Step 3: Authentication and Anomaly Detection
  if Authentication is successful **and** No malicious behavior detected **then**
    RSU and V_i establish Session Key SK.
    Service granted to V_i.
  else
    {Condition: PUF mismatch, OBU compromise, or protocol violation}
    RSU identifies V_i as a malicious/compromised vehicle.
    $RSU \rightarrow TA : \{VID_i, \text{Revocation\_Request}\}$
  end if
end if
Step 4: RL Update and Distribution (by TA)
if TA receives Revocation Request for VID_i **then**
  TA updates Revocation List: $RL \leftarrow RL \cup \{VID_i\}$.
  TA broadcasts the updated RL to all $RSUs$ in the network.
end if

5. Security Analysis

In this section, we analyze the robustness and mutual authentication guarantees of the proposed scheme against various attacks based on informal analysis, the RoR model, BAN logic, and AVISPA.

5.1. Informal Analysis

5.1.1. Replay Attack

Under the Dolev–Yao model, an adversary A can intercept, store, and replay previously transmitted messages over the public channel. For instance, A may replay the message $\{TID_i, c_i, V_1, V_2, N_1, TS_1\}$ in a different session in an attempt to impersonate V_i. However, the protocol binds all authentication tokens to fresh timestamps and session-specific nonces. In particular, $TID_i = VID_i \oplus h(k_i \| TS_1)$, $V_1 = h(VID_i \| n_i \| n_1 \| k_i \| TS_1)$, and $N_1 = n_1 \oplus h(VID_i \| n_i \| k_i \| TS_1)$. Therefore, TID_i, V_1, and N_1 are tightly coupled with the session timestamp TS_1 and nonce n_1. If A replays an old message, the freshness check of TS_1 fails. Even if A attempts to manipulate TS_1, the hash bindings involving k_i, n_i, and VID_i will not be satisfied, causing verification failure. Similarly, later messages such as $\{RID_l, N_2, V_4, V_5, TS_3\}$ and $\{RID_l, UID_j, V_5, V_6, TS_3, TS_4\}$ are also bound to fresh timestamps and session-dependent secrets. Hence, previously captured messages cannot be reused in subsequent sessions, and the protocol is secure against replay attacks.

5.1.2. MITM Attack

An adversary A may attempt a man-in-the-middle attack by intercepting and modifying the exchanged messages among V_i, U_j, and R_l. However, mutual authentication and session key establishment ultimately depend on the MLWE-derived shared secret k_i. Specifically, V_i generates a random message m_i and computes $k_i = h(m_i)$, transmitting the ciphertext $c_i = (u_i, v_i)$. The RSU R_l recovers $m_i^* = Decode(v_i - u_i^T \cdot s_l^*)$ using its private key s_l, and then computes $k_i^* = h(m_i^*)$. The session key is derived as $SK = h(VID_i^* \| UID_j \| RID_l \| n_1^* \| k_i^* \| TS_3)$. Without the legitimate private key s_l, an adversary cannot recover m_i or derive k_i. Any modification of c_i results in an invalid k_i, which consequently invalidates SK and the authentication tokens V_4, V_5, and V_6. Since forging a consistent session requires solving the underlying MLWE problem or recovering s_l, which is computationally infeasible, the protocol is secure against MITM attacks.

5.1.3. Insider Attack

A malicious insider who has completed registration may attempt to exploit stored registration information together with observed protocol messages to derive authentication credentials or session keys. In the proposed protocol, the authentication parameter n_i is protected by the vehicle's PUF. It is reconstructed during authentication as $RE_i^* = PUF(CH_i)$ and $n_i^* = Z_i^2 \oplus h(RE_i^* \| S_i)$. Since RE_i^* cannot be reproduced without access to the physical PUF device, n_i cannot be reconstructed from registration records alone. Furthermore, the session secret $k_i = h(m_i)$ is derived from m_i, which is encapsulated in $c_i = (u_i, v_i)$ and recoverable only through MLWE decapsulation using the RSU private key s_l. Without s_l, an insider cannot obtain m_i or k_i. Because the session key $SK = h(VID_i^* \| UID_j \| RID_l \| n_1^* \| k_i^* \| TS_3)$ depends on both n_i-related values and k_i, an insider observing protocol messages cannot derive a valid session key. Therefore, the protocol is secure against insider attacks.

5.1.4. Privileged Insider Attack

A privileged insider with access to stored registration data such as $\{RID_l, P_l, UID_j, Z_j^1, VID_i\}$ may attempt to exploit this information to impersonate entities or derive session secrets. However, these stored parameters do not include runtime secrets or PUF-derived responses. The authentication parameter n_i depends on $RE_i^* = PUF(CH_i)$ and $n_i^* = Z_i^2 \oplus h(RE_i^* \| S_i)$, which require access to the legitimate physical PUF. Similarly, the session secret $k_i = h(m_i)$ is derived from m_i, encapsulated in $c_i = (u_i, v_i)$ and recoverable only through MLWE decapsulation using the private key s_l. Since s_l is not stored in registration records and cannot be derived from public information, k_i, and consequently

SK, cannot be computed. Therefore, even a privileged insider with access to registration data cannot reconstruct authentication tokens or establish a valid session key. The protocol is secure against privileged insider attacks.

5.1.5. Vehicle Impersonation Attack

An adversary may attempt to impersonate a vehicle by forging the message $\{TID_i, c_i, V_1, V_2, N_1, TS_1\}$. However, such an attempt cannot succeed due to the cryptographic dependencies embedded in these values. The verifier V_1 is computed as $V_1 = h(VID_i \| n_i \| n_1 \| k_i \| TS_1)$, where n_i is protected by the vehicle's PUF and recovered through $RE_i^* = PUF(CH_i)$ and $n_i^* = Z_i^2 \oplus h(RE_i^* \| S_i)$. Since the adversary cannot reproduce the PUF response RE_i^*, it cannot derive the correct n_i. Furthermore, TID_i is generated as $TID_i = VID_i \oplus h(k_i \| TS_1)$, which binds the true identity VID_i to the session secret k_i. Without knowledge of VID_i and k_i, the adversary cannot construct a valid TID_i. Because both n_i and VID_i remain unknown to the adversary, it is infeasible to compute a valid V_1 or related authentication tokens. Any forged message will therefore fail the verification procedures, and the proposed protocol is secure against vehicle impersonation attacks.

5.1.6. UAV Impersonation Attack

An adversary may attempt to impersonate a UAV by forging the message $\{TID_i, UID_j, c_i, V_1, V_3, N_1, TS_1, TS_2\}$. However, the computation of V_3 requires the value Z_j^{2*}, which is derived from the PUF response $RE_j^* = PUF(CH_j)$. Specifically, $Z_j^{1*} = h(UID_j \| RE_j^*)$ and $Z_j^{2*} = Z_j^4 \oplus Z_j^{1*}$, and V_3 is computed as $V_3 = h(TID_i \| UID_j \| V_1 \| Z_j^{2*} \| TS_2)$. Since the adversary cannot obtain the legitimate PUF response RE_j^*, it cannot construct a valid Z_j^{2*} and consequently cannot generate a correct V_3. Therefore, any forged UAV message will fail the RSU's verification, and the proposed protocol is secure against UAV impersonation attacks.

5.1.7. RSU Impersonation Attack

An adversary may attempt to impersonate an RSU by forging the message $\{RID_l, N_2, V_4, V_5, TS_3\}$. The values V_4 and N_2 depend on Z_j^1, which can only be derived through either the UAV's PUF or the RSU's PUF. Moreover, computing V_5 requires knowledge of the session key SK, and the construction of SK depends on k_i, which is derived from n_i, obtained via the vehicle or RSU PUF, and the RSU's private key s_l. Since s_l is protected by the RSU's PUF, the adversary cannot recover k_i or SK. Therefore, the proposed protocol is secure against RSU impersonation attacks.

5.1.8. Vehicle Theft Attack

If an adversary physically steals a vehicle, they may extract stored data $\{VID_i, CH_i, Z_i^2, S_i\}$ through invasive or exhaustive extraction techniques. However, the long-term secret information required for authentication remains protected by the vehicle's PUF. In particular, during authentication the vehicle reconstructs $RE_i^* = PUF(CH_i)$ and derives $n_i^* = Z_i^2 \oplus h(RE_i^* \| S_i)$ (and similarly recovers s_i^* from $S_i \oplus RE_i^*$). Even if CH_i, Z_i^2, and S_i are exposed, an adversary cannot reproduce the correct PUF response RE_i^* without access to the genuine unclonable PUF, and thus cannot reconstruct valid n_i (or related authentication materials). Consequently, the extracted data cannot be exploited to compute valid authenticators such as $V_1 = h(VID_i \| n_i \| n_1 \| k_i \| TS_1)$ or to impersonate V_i in the AKE phase. Therefore, the proposed protocol is secure against vehicle theft attacks.

5.1.9. UAV Capture Attack

An adversary may physically capture a UAV and extract the stored items $\{UID_j, CH_j, Z_j^4\}$. However, the captured data alone is insufficient to forge valid UAV-side authentica-

tion tokens. In the proposed protocol, the UAV computes $RE_j^* = PUF(CH_j)$ and derives $Z_j^{1*} = h(UID_j \| RE_j^*)$, then obtains $Z_j^{2*} = Z_j^4 \oplus Z_j^{1*}$, which is required to generate $V_3 = h(TID_i \| UID_j \| V_1 \| Z_j^{2*} \| TS_2)$. Without the genuine PUF response RE_j^*, an adversary cannot compute Z_j^{1*} and thus cannot recover Z_j^{2*} from Z_j^4. As a result, any attempt to forge $\{TID_i, UID_j, c_i, V_1, V_3, N_1, TS_1, TS_2\}$ will fail the RSU verification of V_3. Therefore, the proposed protocol is resistant to UAV capture attacks.

5.1.10. RSU Table Leakage Attack

An adversary may attempt to extract the table stored in the RSU, i.e., $\{RID_l, CH_l, P_l, S_l, UID_j, Z_j^3, VID_i, Z_i^1\}$. Nevertheless, leaking these stored parameters does not enable the adversary to reconstruct the secret values required for authentication or to derive the session key. In the AKE phase, the RSU derives PUF-dependent secrets by computing $RE_l^* = PUF(CH_l)$ and recovering $s_l^* = S_l \oplus RE_l^*$, and it further computes $Z_j^{2**} = h(UID_j \| n_j^* \| RE_l^* \| s_l^*)$ to verify V_3 via $V_3^* = h(TID_i \| UID_j \| V_1 \| Z_j^{2**} \| TS_2)$. Moreover, the RSU must decapsulate the MLWE ciphertext by computing $m_i^* = Decode(v_i - u_i^T \cdot s_l^*)$ and $k_i^* = h(m_i^*)$, which are subsequently used to derive $SK = h(VID_i^* \| UID_j \| RID_l \| n_1^* \| k_i^* \| TS_3)$. Since the critical PUF response RE_l^* cannot be derived from CH_l alone and the private key material s_l^* cannot be reconstructed without RE_l^*, the adversary cannot compute valid authentication tokens or the session key using only the leaked table information. Therefore, the proposed scheme is secure against RSU table leakage attacks.

5.1.11. Session Key Disclosure Attack

An adversary may attempt to recover the session key by combining previously obtained information with values observed over the public channel. In the proposed protocol, the session key is computed as $SK = h(VID_i^* \| UID_j \| RID_l \| n_1^* \| k_i^* \| TS_3)$, where the critical component k_i^* is established through the MLWE-based KEM. Specifically, $k_i^* = h(m_i^*)$ and $m_i^* = Decode(v_i - u_i^T \cdot s_l^*)$, which requires the RSU's private key s_l^* to decapsulate the ciphertext $c_i = (u_i, v_i)$. Since s_l^* is protected via the RSU's PUF by $s_l^* = S_l \oplus RE_l^*$ with $RE_l^* = PUF(CH_l)$, an adversary observing c_i and public transcripts cannot reconstruct s_l^* and thus cannot obtain m_i^* or k_i^*. Furthermore, n_1^* is also entangled with the authenticated transcript through $N_1 = n_1 \oplus h(VID_i \| n_i \| k_i \| TS_1)$ and subsequent verifications, preventing an attacker from substituting values to derive a consistent SK. Therefore, under the assumed security of the MLWE problem and the unclonability of the PUF, neither passive observation nor combined attacks enable session key disclosure.

5.1.12. Key-Compromise Impersonation Attack

Consider the compromise of vehicle-side secrets such as n_i or s_i belonging to V_i. An adversary may attempt to use these leaked values to impersonate an RSU or UAV toward V_i, or to forge valid authentication messages. However, establishing a valid session still requires the computation of k_i, which is derived as $k_i = h(m_i)$, where m_i is encapsulated in the MLWE ciphertext $c_i = (u_i, v_i)$. Recovering m_i requires decapsulation using the RSU private key s_l via $m_i^* = Decode(v_i - u_i^T \cdot s_l^*)$. Since s_l^* is protected through the RSU's PUF by $s_l^* = S_l \oplus RE_l^*$ with $RE_l^* = PUF(CH_l)$, an adversary cannot derive k_i without access to the legitimate RSU device. Moreover, the session key is computed as $SK = h(VID_i^* \| UID_j \| RID_l \| n_1^* \| k_i^* \| TS_3)$, which depends on k_i. Therefore, even if n_i or s_i are compromised, the adversary cannot impersonate another entity toward V_i without knowledge of k_i. Hence, under the assumption that PUF-protected private keys remain secure, the protocol is resistant to KCI attacks.

5.1.13. Anonymity

In the proposed protocol, the vehicle communicates using a temporary identifier TID_i derived as $TID_i = VID_i \oplus h(k_i \| TS_1)$. Since k_i is a session-dependent secret established via the MLWE-based KEM and known only to V_i and R_l, the real identity VID_i is never transmitted over the public channel. Because recovering VID_i from TID_i requires knowledge of k_i, which in turn depends on the RSU's private key, external observers cannot obtain the true identity of the vehicle. Therefore, the protocol provides anonymity for participating vehicles.

5.1.14. Untraceability

The pseudonym TID_i is generated using session-specific values k_i and TS_1, both of which are freshly generated in each session. Since k_i changes due to new randomness in m_i and the MLWE encapsulation, and TS_1 varies per session, TID_i is unlinkable across different sessions. Consequently, an external adversary observing multiple protocol executions cannot correlate different TID_i values to the same VID_i. Therefore, the proposed protocol achieves untraceability for vehicular communications.

5.1.15. Conditional Traceability

Although a fresh pseudonym TID_i is generated in each session to ensure anonymity and unlinkability, the RSU retains the capability to trace a pseudonym to the real identity when necessary. Since the RSU participates in the MLWE decapsulation process to recover m_i^* and compute k_i^*, it can derive $VID_i^* = TID_i \oplus h(k_i^* \| TS_1)$ during authentication. Thus, in the event of malicious behavior, the RSU can correlate authentication transcripts with stored secret information and recover the true identity of the vehicle. This mechanism enables conditional traceability while preserving privacy under normal operation.

5.1.16. Perfect Forward Secrecy

All session-specific secrets in the AKE phase are freshly generated. The value m_i is randomly chosen in each session, leading to a fresh $k_i = h(m_i)$. The nonce n_1 and timestamps such as TS_3 are also newly generated. Since the session key $SK = h(VID_i^* \| UID_j \| RID_l \| n_1^* \| k_i^* \| TS_3)$ depends on fresh randomness and on k_i, which is derived from the MLWE encapsulation process, compromise of long-term stored data or previous session keys does not enable an adversary to derive future session keys. Therefore, the protocol achieves perfect forward secrecy under the MLWE hardness assumption.

5.1.17. Key Freshness

The session key is computed as $SK = h(VID_i^* \| UID_j \| RID_l \| n_1^* \| k_i^* \| TS_3)$, where n_1^*, k_i^*, and TS_3 are session-dependent values. The nonce n_1 is randomly generated in each execution, and k_i is derived from a freshly chosen m_i encapsulated through MLWE. Because these inputs are independently regenerated in every session, each instance of SK is independent from previous session keys. This prevents key reuse and ensures that compromise of one session does not affect others. Hence, the protocol guarantees key freshness.

5.1.18. Mutual Authentication

Three-party mutual authentication is achieved through the verification of hash-based authenticators. The RSU authenticates V_i and U_j by verifying V_1 and V_3. The UAV authenticates V_i and R_l through verification of V_2 and V_4. Finally, the vehicle authenticates U_j and R_l by verifying V_5 and V_6. Each authentication token is bound to session-specific nonces and timestamps, such as $V_1 = h(VID_i \| n_i \| n_1 \| k_i \| TS_1)$ and $V_6 = h(TID_i \| UID_j \| n_1 \| V_5 \| TS_4)$. Long-term secrets involved in these computations are protected by PUF-derived responses

and private keys. Consequently, only legitimate entities can produce valid authenticators, and the protocol ensures secure three-party mutual authentication.

5.2. Formal Analysis Using AVISPA

To formally verify the proposed protocol, we employ AVISPA [38], a widely used tool for security evaluation. AVISPA is a well-known simulation framework for verifying the security of internet protocols and applications, and it is particularly effective in assessing vulnerabilities such as replay attacks and MITM attacks. The tool uses code written in the high-level protocol specification language (HLPSL) and supports four backend verification engines: the constraint-logic-based attack searcher (CL-AtSe), the on-the-fly model checker (OFMC), the SAT-based model checker (SATMC), and the tree automata-based protocol analyzer (TA4SP). For security property evaluation, HLPSL code is first translated into an intermediate format using the HLPSL2IF translator, after which the backend models perform formal verification.

To validate the security properties of the proposed protocol, the communication procedures described in Section 4 were implemented in HLPSL. Each participating entity (vehicle, UAV, and RSU) was modeled according to its defined operations in the AKE process. The corresponding HLPSL code for each entity is presented in Figure 8.

In this paper, the OFMC and CL-AtSe backend modules were used to simulate the proposed scheme. Figure 9 presents the simulation results, demonstrating that the outputs of both the OFMC and CL-AtSe backend models are classified as SAFE. Therefore, the proposed scheme is capable of resisting replay attacks and MITM attacks.

5.3. Formal Analysis Using the RoR Model

In this paper, we prove the semantic security of the proposed protocol using the RoR model [39]. This model has been widely adopted in prior studies [22,40] for analyzing the security of authentication protocols. In the RoR model, an adversary A interacts with protocol instances and attempts to distinguish the session key SK from a random value. Each participating instance is denoted as p^t, where p represents the entity identifier (e.g., V_i, U_j, R_k) and t denotes the instance index. Under the RoR framework, the adversary A can perform both active and passive attacks by issuing a set of queries to reveal information about the session key, which are summarized in Table 3.

Table 3. Queries and description in RoR model.

Query	Description
$Execute(p_{V_i}^{t_1}, p_{U_j}^{t_2}, p_{R_l}^{t_3})$	Simulates passive eavesdropping, where A collects all exchanged messages during the protocol run.
$Corrupt(p_{V_i}^{t_1})$	Grants A access to the internal memory of the vehicle V_i, exposing stored parameters such as $\{VID_i, CH_i, Z_i^2, S_i\}$.
$Send(p^t, M)$	Models active attacks by letting A send forged messages M to any participant instance and observe the output.
$Test(\Pi^*)$	Used once to challenge the security of the protocol. If $c = 1$, the real session key SK is returned; if $c = 0$, a random string of the same length is returned. A must guess the value of c.

```
role
role_TA(V:agent,U:agent,R:agent,TA:agent,Pl:public_key,UIDj:text,RIDl:text,Key_set_U_TA:(symmetric_key)
set,Key_set_TA_U:(symmetric_key)
set,Key_set_V_TA:(symmetric_key)
set,Key_set_TA_R:(symmetric_key)
set,Key_set_R_TA:(symmetric_key)
set,Key_set_TA_V:(symmetric_key)
set,SND,RCV:channel(dy))
played_by TA
def=
local
State:nat,CHl:text,NNj:text,Zj1:text,Zj2:text,VIDi:text,NNi:text,Key_10:symmetric_key,Key_9:symmetric_key,Key_8:symmetric_key,Key_7:symmetric_key,Key_6:symmetric_key,Key_5:symmetric_key,Key_4:symmetric_key,Key_3:symmetric_key,Key_2:symmetric_key,Key_1:symmetric_key
init
        State := 0
transition
%%% RSU registration
1. State=0 /\ in(Key_1', Key_set_R_TA)
/\ RCV({RIDl'.Pl}_Key_1') =|> State':=1
/\ Key_set_R_TA':=delete(Key_1', Key_set_R_TA)
/\ CHl':=new() /\ Key_2':=new()
/\ Key_set_TA_R':=cons(Key_2', Key_set_TA_R)
/\ SND({CHl'}_Key_2')

%%% UAV registration
1. State=1 /\ in(Key_3', Key_set_U_TA)
/\ RCV({UIDj'.Zj1'}_Key_3') =|> State':=2
/\ Key_set_U_TA':=delete(Key_3', Key_set_U_TA)
/\ NNj':=new() /\ Key_4':=new()
/\ Key_set_TA_R':=cons(Key_4', Key_set_TA_R)
/\ SND({UIDj'.Zj1'.NNj'}_Key_4')
3. State=2 /\ in(Key_5', Key_set_R_TA)
/\ RCV({Zj2'}_Key_5') =|> State':=3
/\ Key_set_R_TA':=delete(Key_5', Key_set_R_TA)
/\ Key_6':=new()
/\ Key_set_TA_U':=cons(Key_6', Key_set_TA_U)
/\ SND({Zj2'}_Key_6')

%%% Vehicle registration
1. State=3 /\ in(Key_7', Key_set_V_TA)
/\ RCV({VIDi'}_Key_7') =|> State':=4
/\ Key_set_V_TA':=delete(Key_7', Key_set_V_TA)
/\ NNi':=new() /\ Key_8':=new()
/\ Key_set_TA_R':=cons(Key_8', Key_set_TA_R)
/\ SND({VIDi'.NNi'}_Key_8')
3. State=4 /\ in(Key_9', Key_set_R_TA)
/\ RCV({NNi}_Key_9') =|> State':=5
/\ Key_set_R_TA':=delete(Key_9', Key_set_R_TA)
/\ Key_10':=new()
/\ Key_set_TA_V':=cons(Key_10', Key_set_TA_V)
/\ SND({NNi}_Key_10')

end role
```

(a) TA

```
role
role_R(V:agent,U:agent,R:agent,TA:agent,Pl:public_key,UIDj:text,RIDl:text,Key_set_TA_R:(symmetric_key)
set,Key_set_R_TA:(symmetric_key)
set,SND,RCV:channel(dy))
played_by R
def=
local
State:nat,SSl,REl,Sl,Zj3,Zi1,UUi,NN1,VVi,KKi,SK,CHl:text,NNj:text,Zj1:text,Zj2:text,VIDi:text,NNi:text,TS2:text,N1:text,V1:text,TIDi:text,CCi:text,V3:text,TS1:text,TS3:text,V4:text,N2:text,V5:text,Key_6:symmetric_key,Key_5:symmetric_key,Key_4:symmetric_key,Key_3:symmetric_key,Key_2:symmetric_key,Key_1:symmetric_key
init
        State := 0
transition
%%% RSU registration
1. State=0 /\ RCV(start) =|> State':=1
/\ Key_1':=new()
/\ Key_set_R_TA':=cons(Key_1', Key_set_R_TA)
/\ RIDl':= new() /\ SSl':=new() /\ SND({RIDl'.Pl}_Key_1')
2. State=1 /\ in(Key_2', Key_set_TA_R)
/\ RCV({CHl'}_Key_2') =|> State':=2 /\ REl':=puf(CHl)
/\ Sl':=xor(SSl, REl')
/\ Key_set_TA_R':=delete(Key_2', Key_set_TA_R)

%%% UAV registration
2. State=2 /\ in(Key_3', Key_set_TA_R)
/\ RCV({UIDj'.Zj1'.NNj'}_Key_3') =|> State':=3
/\ Key_set_TA_R':=delete(Key_3', Key_set_TA_R)
/\ REl':=puf(CHl) /\ Zj2':=h(UIDj', NNj', REl', SSl)
/\ Zj3':=xor((NNj'.Zj1'),h(REl', SSl)) /\ Key_4':=new()
/\ Key_set_R_TA':=cons(Key_4', Key_set_R_TA)
/\ SND({Zj2'}_Key_4')

%%% Vehicle registration
2. State=3 /\ in(Key_5', Key_set_TA_R)
/\ RCV({VIDi'.NNi'}_Key_5') =|> State':=4
/\ Key_set_TA_R':=delete(Key_5', Key_set_TA_R)
/\ Key_6':=new()
/\ Key_set_R_TA':=cons(Key_6', Key_set_R_TA)
/\ REl':=puf(CHl) /\ Zi1':=xor(NNi',h(VIDi', REl'))
/\ SND({NNi'}_Key_6')

%%% Authentication and key exchange
6. State=4
/\ RCV(TIDi'.UIDj.{UUi'.VVi'}_Pl.V1'.V3'.N1'.TS1'.TS2') =|> State':=5
/\ secret(UUi', sec_1,{V,R}) /\ secret(VVi', sec_2,{V,R})
/\ witness(R,U,auth_9,V3') /\ witness(R,V,auth_8,V1')
/\ KKi':=kem(UUi', VVi') /\ REl':=puf(CHl)
/\ VIDi':=xor(TIDi',h(KKi',TS1'))
/\ NNi':= xor(Zi1,h(VIDi', REl'))
/\ NN1':= xor(N1',h(VIDi', NNi', NN1', KKi', TS1')) /\ TS3':=new()
/\ SK':=h(VIDi',UIDj,RIDl, NN1', KKi', TS3')
/\ N2':=xor(NN1',h(Zj1,TS3'))
/\ V4':=h(RIDl,UIDj,Zj1,NN1',TS3')
/\ V5':=h(RIDl,VIDi',SK', TS3')
/\ SND(RIDl.N2'.V4'.V5'.TS3')

end role
```

(b) RSU

(**A**) HLPSL code (TA, RSU)

```
role
role_U(V:agent,U:agent,R:agent,TA:agent,Pl:public_key,
UIDj:text,RIDl:text,Key_set_U_TA:(symmetric_key)
set,Key_set_TA_U:(symmetric_key)
set,SND,RCV:channel(dy))
played_by U
def=
        local

        State:nat,CHj,REj,Zj4,UUi,VVi,NN1,Zj1:text,
Zj2:text,V2:text,TS2:text,N1:text,V1:text,TIDi:text,CCi:text,V3:text,TS1:text,V4:text,N2:text,TS4:text,V6:text,V5:text,TS3:text,Key_2:symmetric_key,Key_1:symmetric_key
        init
                State := 0
        transition
%%% UAV registration
1. State=0 /\ RCV(start) =|> State':=1
/\ UIDj':=new()
/\ CHj':=new()
/\ REj':=puf(CHj')
/\ Zj1':=h(UIDj',REj')
/\ Key_1':=new()
/\ Key_set_U_TA':=cons(Key_1', Key_set_U_TA)
/\ SND({UIDj'.Zj1'}_Key_1')

4. State=1 /\ in(Key_2', Key_set_TA_U)
/\ RCV({Zj2'}_Key_2') =|> State':=2
/\ Zj4':= xor(Zj1,Zj2')
/\ Key_set_TA_U':=delete(Key_2', Key_set_TA_U)

%%% Authentication and key exchange
5. State=2 /\ RCV(TIDi'.{UUi'.VVi'}_Pl.V1'.V2'.N1'.TS1')
=|> State':=3
/\ secret(VVi', sec_2,{V,R})
/\ witness(U,V,auth_6,V2')
/\ secret(UUi', sec_1,{V,R})
/\ TS2':=new()
/\ REj':=puf(CHj)
/\ Zj1':=h(UIDj,REj')
/\ Zj2':=xor(Zj4,Zj1')
/\ V3':=h(TIDi',UIDj,V1, Zj2', TS2')
/\ SND(TIDi'.UIDj.{UUi'.VVi'}_Pl.V1'.V3'.N1'.TS1'.TS2')


7. State=3 /\ RCV(RIDl.N2'.V4'.V5'.TS3') =|> State':=4
/\ witness(U,R,auth_7,V4')
/\ TS4':=new()
/\ NN1':=xor(N2',h(Zj1,TS3'))
/\ V6':=h(TIDi,UIDj,NN1', V5', TS4')
/\ SND(RIDl.UIDj.V5'.V6'.TS3'.TS4')
end role
```

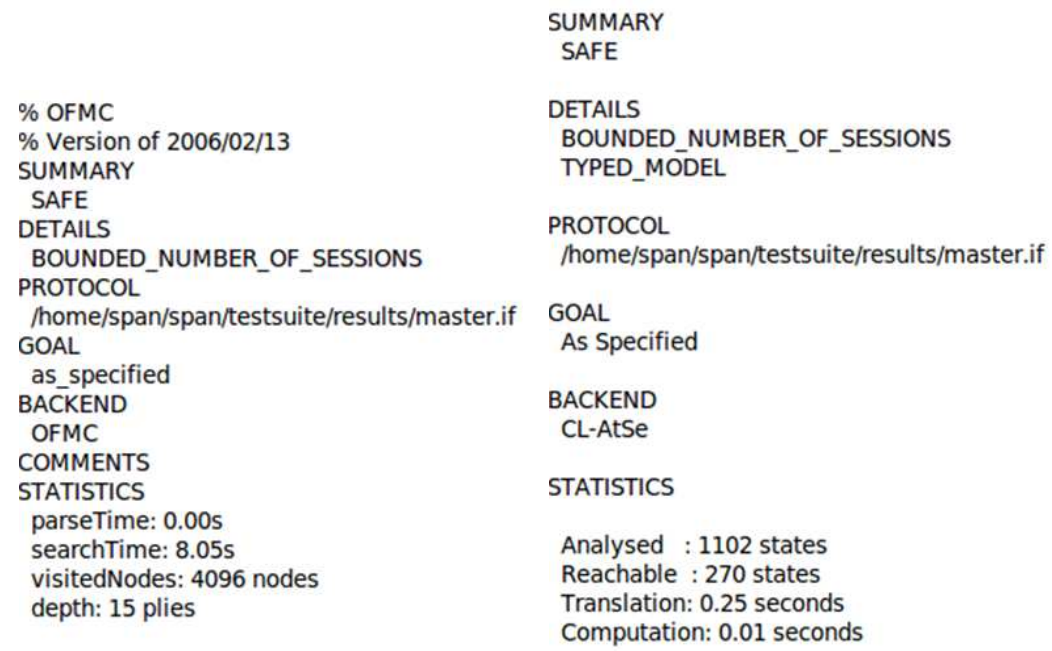
(c) UAV

```
role
role_V(V:agent,U:agent,R:agent,TA:agent,Pl:public_key,
UIDj:text,RIDl:text,Key_set_V_TA:(symmetric_key)
set,Key_set_TA_V:(symmetric_key)
set,SND,RCV:channel(dy))
played_by V
def=
        local

        State:nat,SSi,CHi,REi,Zi2,Si,NN1,UUi,VVi,KKi,SK,VIDi:text,NNi:text,TS1:text,V2:text,CCi:text,TIDi:text,V1:text,N1:text,TS4:text,V6:text,V5:text,TS3:text,Key_2:symmetric_key,Key_1:symmetric_key
        init
                State := 0
        transition
%%% Vehicle registration
1. State=0 /\ RCV(start) =|> State':=1
/\ VIDi':=new()
/\ Key_1':=new()
/\ Key_set_V_TA':=cons(Key_1', Key_set_V_TA)
/\ SND({VIDi'}_Key_1')

4. State=1 /\ in(Key_2', Key_set_TA_V)
/\ RCV({NNi'}_Key_2') =|> State':=2
/\ Key_set_TA_V':=delete(Key_2', Key_set_TA_V)
/\ SSi':=new()
/\ CHi':=new()
/\ REi':=puf(CHi')
/\ Zi2':=xor(NNi',h(REi', SSi'))
/\ Si':= xor(SSi', REi')

%%% Authentication and key exchange
/\ NN1':=new()
/\ TS1':=new()
/\ UUi':=new()
/\ VVi':=new()
/\ KKi':=kem(UUi', VVi')
/\ TIDi':=xor(VIDi,h(KKi', TS1'))
/\ V1':=h(VIDi,NNi', KKi', TS1')
/\ V2':=h(TIDi', NNi', KKi', TS1')
/\ N1':=xor(NN1',h(VIDi, NNi', KKi', TS1'))
/\ secret(UUi', sec_1,{V,R})
/\ secret(VVi', sec_2,{V,R})
/\ SND(TIDi'.{UUi'.VVi'}_Pl.V1'.V2'.N1'.TS1')


8. State=2 /\ RCV(RIDl.UIDj.V5'.V6'.TS3'.TS4') =|>
State':=3
/\ witness(V,R,auth_5,V5')
/\ V6':=h(TIDi,UIDj,NN1,V5,TS4)
/\ SK':=h(VIDi,UIDj,RIDl,NN1,KKi,TS3')
/\ V5':=h(RIDl,VIDi,SK', TS3')


end role
```

(d) Vehicle

(**B**) HLPSL code (UAV, Vehicle)

Figure 8. HLPSL specification of the proposed protocol.



(**a**) OFMC (**b**) CL-AtSe

Figure 9. Result of AVISPA simulation.

In this paper, $\mid PUF \mid$ and $\mid Hash \mid$ denote the output ranges of the PUF $PUF(\cdot)$ and the hash function $H(\cdot)$, respectively. In addition, q_p and q_h represent the number of PUF and hash queries performed by the adversary A.

Proof. Let $Adv(A)$ denote the advantage of adversary A in distinguishing the session key. The security is analyzed through a sequence of games G_0 to G_4 as follows:

- G_0: In the initial game, A has no information. Therefore,

$$Adv(A) = Adv_{G_1}(A) \tag{1}$$

- G_1: A issues Execute queries and can observe the complete transcripts. Since $SK = h(VID_i \mid\mid UID_j \mid\mid RID_l \mid\mid n_1 \mid\mid k_i \mid\mid TS_3)$ and k_i is derived through an IND-CCA-secure MLWE-based KEM, A cannot infer SK through passive observation alone. Thus,

$$Adv_{G_0}(A) = Adv_{G_1}(A) \tag{2}$$

- G_2: A is allowed to perform Send and Hash queries. The only way to reveal SK is to find a collision in the hash function $h(\cdot)$. Using the birthday bound,

$$\mid Adv_{G_2}(A) - Adv_{G_1}(A) \mid \leq \frac{q_h^2}{2 \mid Hash \mid} \tag{3}$$

- G_3: A may issue malicious queries and attempt to extract the secret credentials $\{VID_i, CH_i, Z_i^2, S_i\}$ from the vehicle via power analysis attacks. However, note that $N_1 = n_1 \oplus h(VID_i \mid n_i \mid k_i \mid TS_1)$ and $n_i = Z_i^2 \oplus h(RE_i \mid s_i)$. To expose the session key, A must guess n_i using Send and PUF queries. Hence,

$$\mid Adv_{G_3}(A) - Adv_{G_2}(A) \mid \leq \frac{q_p^2}{2 \mid PUF \mid} \tag{4}$$

- G_4: Finally, the success of A depends on recovering k_i by inverting the MLWE-based ciphertext c_i, which is computationally infeasible under the MLWE assumption:

$$| Adv_{G_4}(A) - Adv_{G_3}(A) | \leq Adv_A^{MLWE} \tag{5}$$

By the triangle inequality:

$$\begin{aligned} | Adv_{G_4}(A) - Adv_{G_1}(A) | \leq & | Adv_{G_4}(A) - Adv_{G_3}(A) | + | Adv_{G_3}(A) - Adv_{G_2}(A) | \\ & + | Adv_{G_2}(A) - Adv_{G_1}(A) | \\ & \leq \frac{q_h^2}{2 | Hash |} + \frac{q_p^2}{2 | PUF |} + Adv_A^{MLWE} \end{aligned} \tag{6}$$

Therefore, the overall advantage of the adversary is bounded as

$$Adv(A) \leq \frac{q_h^2}{2 | Hash |} + \frac{q_p^2}{2 | PUF |} + Adv_A^{MLWE} \tag{7}$$

Thus, the proposed protocol is semantically secure under the RoR model, assuming the hardness of the hash function, the PUF, and the MLWE problem. □

5.4. Formal Analysis Using BAN Logic

To demonstrate the correctness of the proposed protocol, we employ BAN logic [41]. BAN logic is a formal analytical method used to verify whether a shared session key can be securely established among communicating entities within a security protocol.

The BAN logic-based analysis proceeds as follows. First, the security goals of the protocol are defined. Then, the protocol messages are transformed into their idealized abstract forms. Subsequently, the required initial assumptions are specified, and the axioms and inference rules of BAN logic are applied to verify whether each communicating party can derive the intended security properties.

Before conducting the BAN logic analysis, we define the notation used throughout the formal reasoning process, along with brief explanations. The primary BAN logic symbols employed in this paper are summarized in Table 4.

Table 4. BAN logic notation.

Notation	Description
ρ_1, ρ_2	Two principals
s_1, s_2	Two statements
$\rho_1 \mid \equiv s_1$	ρ_1 **believes** s_1
$\rho_1 \mid \sim s_1$	ρ_1 once **said** s_1
$\rho_1 \Rightarrow s_1$	ρ_1 **controls** s_1
$\rho_1 \lhd \mu_1$	ρ_1 **receives** s_1
$\#s_1$	s_1 is **fresh**
$(s_1)_K$	s_1 is **encrypted** with K
$\rho_1 \overset{K}{\leftrightarrow} \rho_2$	ρ_1 and ρ_2 have **shared** key K
SK	The session key

5.4.1. BAN Logic Rules

1. Message meaning rule (MMR):

$$\frac{\rho_1 \mid \equiv \rho_1 \overset{K}{\leftrightarrow} \rho_2, \quad \rho_1 \lhd (s_1)_K}{\rho_1 \mid \equiv \rho_2 \mid \sim s_1}$$

2. Nonce verification rule (NVR):

$$\frac{\rho_1 \mid\equiv \#(s_1),\quad \rho_1 \mid\equiv \rho_2 \mid\sim s_1}{\rho_1 \mid\equiv \rho_2 \mid\equiv s_1}$$

3. Jurisdiction rule (JR):

$$\frac{\rho_1 \mid\equiv \rho_2 \mid\Longrightarrow s_1,\quad \rho_1 \mid\equiv \rho_2 \mid\equiv s_1}{\rho_1 \mid\equiv s_1}$$

4. Belief rule (BR):

$$\frac{\rho_1 \mid\equiv (s_1, s_2)}{\rho_1 \mid\equiv s_1}$$

5. Freshness rule (FR):

$$\frac{\rho_1 \mid\equiv \#(s_1)}{\rho_1 \mid\equiv \#(s_1, s_2)}$$

5.4.2. Goals

The authentication goals are to establish that both the vehicle V_i and the base RSU RSU_d^d believe they share a common session key SK, and that each believes the other believes so as well.

Goal 1: $V_i \mid\equiv V_i \overset{SK}{\longleftrightarrow} R_l$

Goal 2: $R_l \mid\equiv V_i \overset{SK}{\longleftrightarrow} R_l$

Goal 3: $V_i \mid\equiv R_l \mid\equiv V_i \overset{SK}{\longleftrightarrow} R_l$

Goal 4: $R_l \mid\equiv V_i \mid\equiv V_i \overset{SK}{\longleftrightarrow} R_l$

5.4.3. Idealized Forms

The idealized message forms are defined as follows. Msg_1 and Msg_2 mean that V_i and R_l send the confidential values. In the proposed protocol, U_j acts as an intermediary between the vehicle V_i and the RSU R_l. Although U_j participates in the mutual authentication process to verify the legitimacy of the communicating parties, it is not a trusted authority and does not contribute to the computation of the session key SK. Therefore, U_j is excluded from the BAN logic derivation, and only V_i and R_l are considered for the formal analysis of the key exchange process.

MSG1: $V_i \rightarrow R_l : \{VID_i, n_1, TS_1\}_{k_i}$

MSG2: $R_l \rightarrow V_i : \{VID_i, n_1, TS_3\}_{k_i}$

5.4.4. Assumptions

A1: $R_l \mid\equiv \#(TS_1)$

A2: $R_l \mid\equiv V_i \overset{k_i}{\longleftrightarrow} R_l$

A3: $R_l \mid\equiv V_i \Rightarrow (V_i \overset{SK}{\longleftrightarrow} R_l)$

A4: $V_i \mid\equiv \#(TS_3)$

A5: $V_i \mid\equiv V_i \overset{k_i}{\longleftrightarrow} R_l)$

A6: $V_i \mid\equiv R_l \Rightarrow (V_i \overset{SK}{\longleftrightarrow} R_l)$

5.4.5. BAN Logic Proof

Proof. The process of analysis using BAN logic includes the detailed steps below.

Step 1: According to Msg_1, S_1 is derived.

$$S_1 : R_l \triangleleft \{VID_i, n_1, TS_1\}_{k_i}$$

Step 2: Applying S_1 and A_2 to the MMR, S_2 is obtained.

$$S_2 : R_l \mid\equiv V_i \mid\sim (VID_i, n_1, TS_1)$$

Step 3: Applying A_1 to the FR, S_3 is obtained.

$$S_3 : R_l \mid\equiv \#(VID_i, n_1, TS_1)$$

Step 4: Applying S_2 and S_3 to the NVR, S_4 is obtained.

$$S_4 : R_l \mid\equiv V_i \mid\equiv (VID_i, n_1, TS_1)$$

Step 5: According to Msg_2, S_5 is obtained.

$$S_5 : V_i \triangleleft \{VID_i, n_1, TS_3\}_{k_i}$$

Step 6: Applying S_5 and A_5 to the MMR, S_6 is obtained.

$$S_6 : V_i \mid\equiv R_l \mid\sim (VID_i, n_1, TS_3)$$

Step 7: Applying A_4 to the FR, S_7 is obtained.

$$S_7 : V_i \mid\equiv \#(VID_i, n_1, TS_3)$$

Step 8: Applying S_6 and S_7 to the NVR, S_8 is obtained.

$$S_8 : V_i \mid\equiv R_l \mid\equiv (VID_i, n_1, TS_3)$$

Step 9: R_l and V_i believe that the other party can compute the session key $SK = h(VID_i \mid\mid UID_j \mid\mid RID_l \mid\mid n_1 \mid\mid k_i \mid\mid TS_3)$. Because SK can be calculated using the shared values,

$$S_9 : R_l \mid\equiv V_i \mid\equiv V_i \overset{SK}{\longleftrightarrow} R_l \textbf{ (Goal 4)}$$
$$S_{10} : V_i \mid\equiv R_l \mid\equiv V_i \overset{SK}{\longleftrightarrow} R_l \textbf{ (Goal 3)}$$

Step 10: Applying S_9 and A_3 to the JR, S_{11} is obtained. And applying S_{10} and A_6 to the JR, S_{12} is obtained.

$$S_{11} : R_l \mid\equiv V_i \overset{SK}{\longleftrightarrow} R_l \textbf{ (Goal 2)}$$
$$S_{12} : V_i \mid\equiv V_i \overset{SK}{\longleftrightarrow} R_l \textbf{ (Goal 1)}$$

□

6. Performance Analysis

6.1. Comparison of Security Features

A comparative analysis was conducted to evaluate the security capabilities of the proposed scheme and other related approaches [18,19,22,42,43]. The results are summarized in Table 5. The comparison shows that the proposed scheme provides stronger and more comprehensive security guarantees than existing protocols.

Table 5. Comparison of security properties across protocols.

Security Property	Choi et al. [22]	Ghosh et al. [42]	Miao et al. [19]	El-Zawawy et al. [18]	Zhang et al. [43]	Proposed
Replay Attack	×	✓	×	✓	✓	✓
MITM Attack	×	×	×	×	×	✓
Insider Attack	✓	✓	✓	×	✓	✓
Vehicle Impersonation	×	×	×	×	×	✓
UAV Impersonation	×	×	×	×	×	✓
RSU Impersonation	✓	×	×	–	–	✓
Privileged Insider Attack	✓	✓	✓	✓	×	✓
Vehicle Theft	×	×	×	×	✓	✓
UAV Capture	×	×	×	×	×	✓
Session Key Disclosure	×	✓	×	×	✓	✓
RSU Table Leakage Attack	×	×	×	×	–	✓
Anonymity	✓	✓	×	✓	×	✓
Untraceability	×	✓	×	×	✓	✓
Session Key Inconsistency	✓	✓	✓	×	✓	✓
Post-Quantum Security	×	×	×	×	×	✓
ID Synchronization Problem	×	✓	×	–	–	✓

✓: guarantees the security feature, ×: does not guarantee the security feature, –: does not consider the security feature

6.2. Computational Cost Analysis

We analyze the computational costs incurred during the AKE phases of the proposed protocol and compare them with those of existing approaches [18,19,22,42,43]. The total computational overhead of each protocol is quantitatively evaluated based on the execution times of the cryptographic primitives employed in their authentication procedures. Table 6 summarizes the measured execution times of the cryptographic primitives under both RSU and UAV/vehicle configurations, including SHA-256 hashing, AES-128 encryption and decryption, ECC multiplication and addition, and Kyber-512 encapsulation and decapsulation operations. To more realistically approximate resource-constrained environments, we constructed a virtualization-based hardware limitation setup using Oracle VM VirtualBox. Two representative configurations were defined as follows:

- **RSU:** 4 GB memory, six processor cores, and a CPU execution cap set to 100%.
- **UAV/Vehicle:** 4 GB memory, four processor cores, and a CPU execution cap set to 40%.

In the UAV/vehicle configuration, the number of processor cores was reduced from six to four and the CPU execution cap was limited to 40%. By combining the reduced core allocation (approximately 67% of the RSU configuration) with the execution cap constraint, the environment was modeled to provide approximately 27% of the effective processing capability of the RSU configuration. This configuration is intended to reflect the fact that vehicular OBUs and UAV onboard platforms typically rely on ARM-based low-power system-on-chip (SoC) architectures, which exhibit lower computational performance than desktop/server-class x86 CPUs in terms of core count, clock frequency, and cache capacity [44–46]. Furthermore, prior VANET and UAV platform studies report that embedded vehicular and UAV boards are subject to size, weight, and power constraints, resulting in substantially lower computational capabilities compared to infrastructure-grade or desktop environments [46–48]. Furthermore, according to Xia et al. [49], the execution time of the fuzzy extractor is considered equivalent to that of elliptic curve point multiplication. Therefore, we assume $T_f = T_{em}$ in both experimental settings for the computational cost analysis.

Table 6. Execution times of cryptographic primitives.

Operation	Symbol	RSU (ms)	UAV/Vehicle (ms)
Hash	T_h	0.0004	0.0011
Fuzzy extractor	T_f	0.0354	0.0956
AES encryption	T_{senc}	0.0001	0.0003
AES decryption	T_{sdec}	0.0001	0.0004
ECC multiplication	T_{em}	0.0354	0.0956
ECC addition	T_{ea}	0.0006	0.0017
Kyber encapsulation	T_{kenc}	0.0274	0.0739
Kyber decapsulation	T_{kdec}	0.0349	0.0943

Table 7 and Figure 10 present the summarized computational costs of each protocol, calculated using the operation times described above. The costs were analyzed based on the cryptographic operations performed by different entities, including the vehicle, UAV, and RSU, and the execution times are aggregated in milliseconds.

Table 7. Comparison cost comparison.

Protocol	Vehicle (ms)	UAV (ms)	RSU (ms)	Total Cost (ms)
Choi et al. [22]	$10T_h + T_f \approx 0.1065$	$8T_h \approx 0.0087$	$14T_h \approx 0.0055$	0.1207
Ghosh et al. [42]	$3T_h + 4T_{em} + T_{ea} \approx 0.3874$	$4T_h + 5T_{em} + T_{ea} \approx 0.4841$	$3T_h + 3T_{em} \approx 0.1074$	0.9789
Miao et al. [19]	$6T_h + 4T_{em} + 4T_{ea} \approx 0.3907$	$4T_h + 3T_{em} \approx 0.2911$	$8T_h + 6T_{em} + T_{senc} + T_{sdec} \approx 0.2159$	0.8977
El-Zawawy et al. [18]	$6T_h + 4T_{em} + 4T_{ea} \approx 0.3959$	$11T_h + 9T_{em} + 8T_{ea} \approx 0.8863$	$6T_h + 2T_{em} + 4T_{ea} \approx 0.0758$	1.3580
Zhang et al. [43]	$5T_h + 2T_{em} + T_{senc} \approx 0.1968$	$4T_h + T_{em} + T_{senc} + T_{sdec} \approx 0.1002$	$7T_h + T_{em} + T_{senc} + 2T_{sdec} \approx 0.0386$	0.3356
Proposed	$8T_h + T_{kenc} \approx 0.0826$	$6T_h \approx 0.0065$	$11T_h + T_{kdec} \approx 0.0987$	0.1878

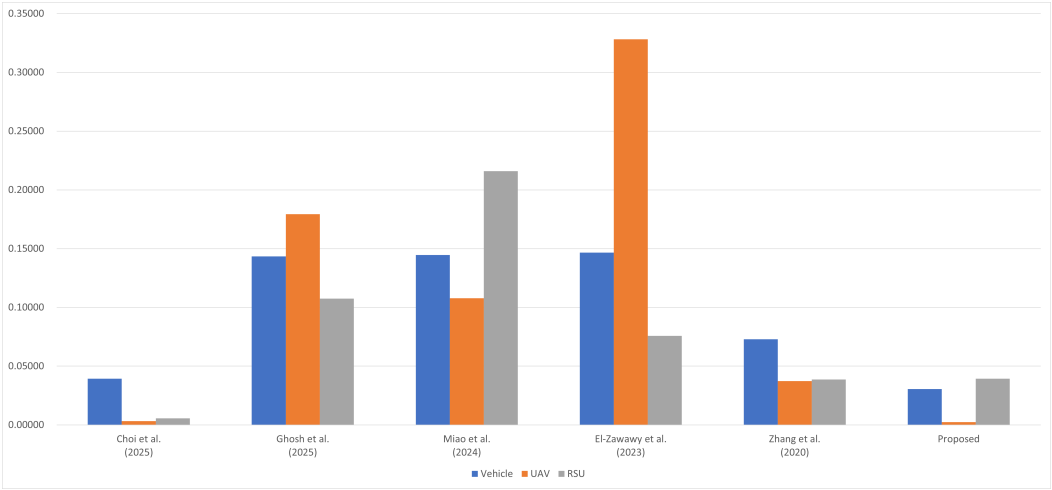


Figure 10. Comparison of computation cost. Refs. [18,19,22,42,43].

The total computational cost of the proposed scheme is approximately 0.07221 ms, achieving up to an 85% reduction compared with the schemes of Ghosh et al. (0.43013 ms), Miao et al. (0.46832 ms), and El-Zawawy et al. (0.55055 ms). This improvement primarily results from the structural design of the proposed protocol, in which mutual authentication and session key establishment are simultaneously achieved through a single pair of MLWE-based KEM encapsulation and decapsulation operations. Unlike existing approaches that rely on multiple ECC operations or repeated symmetric-key encryptions, the proposed method minimizes expensive computations and avoids redundant cryptographic primitives. In particular, the computational cost on the UAV side is only 0.00238 ms, which is the lowest among all the compared schemes, demonstrating high practicality in UAV-assisted VANET environments where UAVs must perform real-time relay and verification tasks. Furthermore, both RSU-side and vehicle-side computation costs remain low, confirming

that the proposed scheme provides a balanced combination of efficiency and real-time performance across all participating entities.

Additionally, scalability with respect to an increasing number of vehicles must be considered. Figure 11 presents a comparison of the total computational cost as the number of vehicles increases. The proposed scheme incurs lower computational overhead than existing approaches, enabling efficient operation even in high-density vehicular environments.

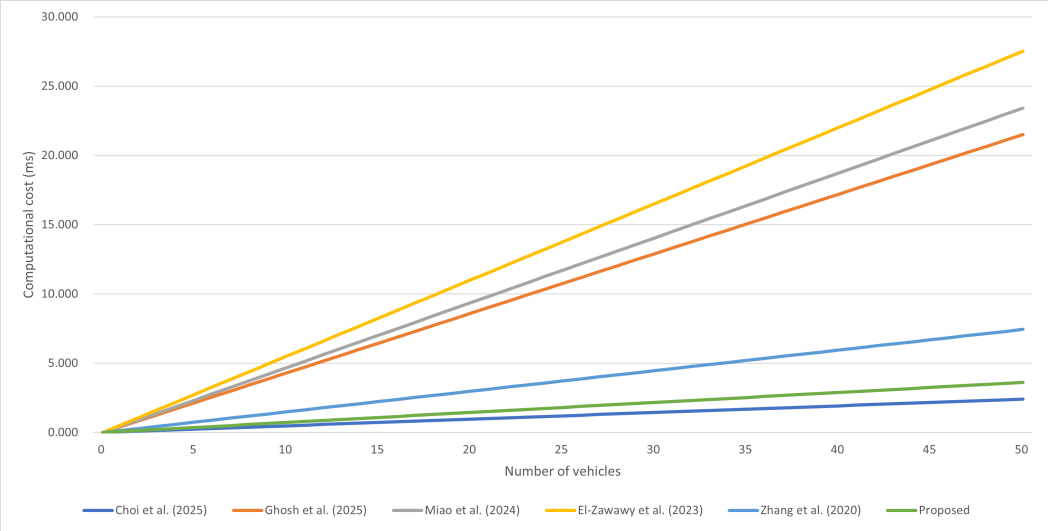


Figure 11. Scalability analysis in terms of the number of vehicles. Refs. [18,19,22,42,43].

6.3. Communication Cost Analysis

This section analyzes the communication overhead of the proposed scheme under bandwidth constraints in practical VANET environments. The vehicular communication setting is assumed to be based on dedicated short-range communications (DSRC), where the physical layers (PHY) and MAC layers comply with the IEEE 802.11p standard. IEEE 802.11p supports data rates ranging from 3 Mbps to 27 Mbps over a 10 MHz channel. The analysis is conducted based on the IEEE 802.11p-based DSRC standard described in [50].

According to the IEEE 802.11 specification, the maximum MAC frame body size is 2304 bytes. In the proposed scheme, the size of each message component is defined as follows: an ID and verification data occupy 32 bytes each, a timestamp requires 4 bytes, and the Crystals-Kyber KEM ciphertext has a size of 768 bytes. Based on these parameters, the message size of $\{TID_i, c_i, V_1, V_2, N_1, TS_1\}$ is 900 bytes, whereas that of $\{TID_i, UID_j, c_i, V_1, V_3, N_1, TS_1, TS_2\}$ is 936 bytes. In addition, the message sizes of $\{RID_l, N_2, V_4, V_5, TS_3\}$ and $\{RID_l, UID_j, V_5, V_6, TS_3, TS_4\}$ are 132 bytes and 136 bytes, respectively. The maximum message size is therefore 936 bytes, which is well below the MAC frame body limit of 2304 bytes. As a result, MAC-layer fragmentation is not required, thereby avoiding additional overhead and retransmission delays.

In terms of transmission latency, the largest message (936 bytes) corresponds to 7488 bits. At a PHY data rate of 3 Mbps, the transmission time is approximately 2.5 ms, while it decreases to approximately 1.25 ms at 6 Mbps. Even when additional protocol overhead, such as MAC headers and PHY preambles, is taken into account, the overall transmission delay remains within the millisecond range.

To reflect realistic traffic density, a medium-density VANET scenario is considered in which 50 vehicles broadcast authentication-related messages at a frequency of 10 Hz. Under this setting, each vehicle generates approximately 936 bytes $\times$ 10 $=$ 9360 bytes per second, corresponding to about 74.9 kbps. The aggregate channel load for 50 vehicles is

therefore approximately 3.74 Mbps, corresponding to about 62% channel occupancy at a 6 Mbps PHY rate and approximately 31% at 12 Mbps.

IEEE 802.11p-based vehicular networks incorporate congestion control mechanisms, such as SAE J2945.1 and ETSI DCC [51], which dynamically adjust transmission rates under increased channel load conditions. Furthermore, considering the packet delivery ratio (PDR) degradation caused by distance and obstacles [52,53], the effective channel load perceived by a specific RSU or UAV is typically lower than the theoretical aggregate load. Therefore, in practical deployments, the channel occupancy is expected to remain below saturation levels. Overall, the communication overhead introduced by the proposed scheme remains within feasible bandwidth limits for medium-density VANET environments.

7. Conclusions

In this paper, we proposed a novel authentication and key exchange protocol that combines MLWE and PUF technologies to address security vulnerabilities in UAV-assisted VANET environments and overcome the limitations of conventional public key-based authentication frameworks. The proposed scheme enhances structural security by treating UAVs as untrusted relay nodes and ensures that a secure session key is established exclusively between vehicles and RSUs. Furthermore, the protocol is designed with lightweight computational operations, taking into account the characteristics of VANETs that require high mobility and real-time communication, thereby achieving both strong security and high efficiency.

Security evaluation demonstrated that the proposed scheme is secure against various attacks, including replay, MITM, impersonation, and insider attacks, validated through informal analysis, RoR model-based formal proof, BAN logic reasoning, and AVISPA-based simulation. In addition, performance analysis revealed that the proposed approach significantly reduces computational overhead compared with existing UAV-assisted VANET authentication protocols, particularly minimizing the burden on UAVs, which confirms its suitability for real-time vehicular communication environments.

Future work will focus on developing an extended security framework that considers load balancing and mobility management in large-scale vehicular networks with cooperative multi-UAV support, as well as optimizing the protocol and conducting further simulation studies for real-world deployment.

Author Contributions: Conceptualization, Y.P.; methodology, H.P.; software, H.P.; validation, H.P.; formal analysis, H.P.; investigation, H.P.; writing—original draft, H.P.; writing—review and editing, Y.P.; supervision, Y.P.; project administration, Y.P.; funding acquisition, Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Bisa Research Grant of Keimyung University in 2025 (Project No: 20250485).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in [AVISPA] [https://github.com/wonny0124/AVISPA/blob/main/AVISPA_simulation, accessed on 17 February 2026].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Shor, P.W. Algorithms for quantum computation: Discrete logarithms and factoring. In Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 20–22 November 1994; pp. 124–134.
2. Grover, L.K. A fast quantum mechanical algorithm for database search. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, Philadelphia, PA, USA, 22–24 May 1996; pp. 212–219.
3. Kumar, M.; Pattnaik, P. Post quantum cryptography (pqc)-an overview. In Proceedings of the 2020 IEEE High Performance Extreme Computing Conference (HPEC), Waltham, MA, USA, 22–24 September 2020; pp. 1–9.
4. Joseph, D.; Misoczki, R.; Manzano, M.; Tricot, J.; Pinuaga, F.D.; Lacombe, O.; Leichenauer, S.; Hidary, J.; Venables, P.; Hansen, R. Transitioning organizations to post-quantum cryptography. *Nature* **2022**, *605*, 237–243. [CrossRef] [PubMed]
5. Alagic, G.; Bros, M.; Ciadoux, P.; Cooper, D.; Dang, Q.; Dang, T.; Kelsey, J.; Lichtinger, J.; Liu, Y.K.; Miller, C.; et al. *Status Report on the Fourth Round of the Nist Post-Quantum Cryptography Standardization Process*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2025.
6. Hasija, T.; Ramkumar, K.; Kaur, A.; Mittal, S.; Singh, B. A survey on nist selected third round candidates for post quantum cryptography. In Proceedings of the 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 22–24 June 2022; pp. 737–743.
7. Park, H.; Park, Y. Formal Security Analysis of the Authentication Protocol in Smart Cities Using AVISPA. In *Proceedings of the International Conference on Computational Science*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 3–17.
8. Park, H.; Son, S.; Park, Y.; Park, Y. Provably Quantum Secure Three-Party Mutual Authentication and Key Exchange Protocol Based on Modular Learning with Error. *Electronics* **2024**, *13*, 3930. [CrossRef]
9. Prajapat, S.; Gautam, D.; Kumar, P.; Jangirala, S.; Das, A.K.; Park, Y.; Lorenz, P. Secure lattice-based aggregate signature scheme for vehicular Ad Hoc networks. *IEEE Trans. Veh. Technol.* **2024**, *73*, 12370–12384. [CrossRef]
10. Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A.K. Design of blockchain-based lightweight V2I handover authentication protocol for VANET. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1346–1358. [CrossRef]
11. Wazid, M.; Singh, J.; Pandey, C.; Sherratt, R.S.; Das, A.K.; Giri, D.; Park, Y. Explainable deep Learning-Enabled malware attack detection for IoT-Enabled intelligent transportation systems. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 7231–7244. [CrossRef]
12. Wazid, M.; Bagga, P.; Das, A.K.; Shetty, S.; Rodrigues, J.J.; Park, Y. AKM-IoV: Authenticated key management protocol in fog computing-based Internet of vehicles deployment. *IEEE Internet Things J.* **2019**, *6*, 8804–8817. [CrossRef]
13. Gautam, D.; Thakur, G.; Kumar, P.; Das, A.K.; Park, Y. Blockchain assisted intra-twin and inter-twin authentication scheme for vehicular digital twin system. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 15002–15015. [CrossRef]
14. Kwon, D.; Son, S.; Kim, M.; Lee, J.; Das, A.K.; Park, Y. A secure self-certified broadcast authentication protocol for intelligent transportation systems in UAV-assisted mobile edge computing environments. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 19004–19017. [CrossRef]
15. Gupta, L.; Jain, R.; Vaszkun, G. Survey of important issues in UAV communication networks. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1123–1152. [CrossRef]
16. Menouar, H.; Guvenc, I.; Akkaya, K.; Uluagac, A.S.; Kadri, A.; Tuncer, A. UAV-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Commun. Mag.* **2017**, *55*, 22–28. [CrossRef]
17. Ng, J.S.; Lim, W.Y.B.; Dai, H.N.; Xiong, Z.; Huang, J.; Niyato, D.; Hua, X.S.; Leung, C.; Miao, C. Joint auction-coalition formation framework for communication-efficient federated learning in UAV-enabled Internet of Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 2326–2344. [CrossRef]
18. El-Zawawy, M.A.; Brighente, A.; Conti, M. Authenticating drone-assisted internet of vehicles using elliptic curve cryptography and blockchain. *IEEE Trans. Netw. Serv. Manag.* **2022**, *20*, 1775–1789. [CrossRef]
19. Miao, J.; Wang, Z.; Ning, X.; Shankar, A.; Maple, C.; Rodrigues, J.J. A UAV-assisted authentication protocol for internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 10286–10297. [CrossRef]
20. Cui, J.; Liu, X.; Zhong, H.; Zhang, J.; Wei, L.; Bolodurina, I.; He, D. A practical and provably secure authentication and key agreement scheme for UAV-assisted VANETs for emergency rescue. *IEEE Trans. Netw. Sci. Eng.* **2023**, *11*, 1454–1468. [CrossRef]
21. Guo, Z.; Cao, J.; Wang, X.; Zhang, Y.; Niu, B.; Li, H. UAVA: Unmanned aerial vehicle assisted vehicular authentication scheme in edge computing networks. *IEEE Internet Things J.* **2024**, *11*, 22091–22106. [CrossRef]
22. Choi, J.; Kwon, D.; Son, S.; Park, Y.; Das, A.K.; Park, Y. A PUF-Based Lightweight Authentication Scheme for UAV-Assisted Internet of Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 13782–13798. [CrossRef]
23. Lin, L.; Shangguan, R.; Ge, H.; Liu, Y.; Zhou, Y.; Zhou, Y. Mutual Identity Authentication Based on Dynamic Identity and Hybrid Encryption for UAV–GCS Communications. *Drones* **2025**, *9*, 422. [CrossRef]
24. Telikani, A.; Sarkar, A.; Du, B.; Santoso, F.; Shen, J.; Yan, J.; Yong, J.; Yap, E. Unmanned aerial vehicle-aided intelligent transportation systems: Vision, challenges, and opportunities. *IEEE Commun. Surv. Tutor.* **2025**, *27*, 3772–3819. [CrossRef]
25. Regev, O. On lattices, learning with errors, random linear codes, and cryptography. In Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, 22–24 May 2005; pp. 84–93.

26. Peikert, C. A decade of lattice cryptography. *Found. Trends Theor. Comput. Sci.* **2016**, *10*, 283–424. [CrossRef]
27. Langlois, A.; Stehlé, D. Worst-case to average-case reductions for module lattices. *Des. Codes Cryptogr.* **2015**, *75*, 565–599. [CrossRef]
28. Avanzi, R.; Bos, J.; Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schanck, J.M.; Schwabe, P.; Seiler, G.; Stehlé, D.; et al. CRYSTALS-Kyber algorithm specifications and supporting documentation. *NIST PQC Round* **2019**, *2*, 1–43.
29. Gao, Y.; Al-Sarawi, S.F.; Abbott, D. Physical unclonable functions. *Nat. Electron.* **2020**, *3*, 81–91. [CrossRef]
30. Yu, S.; Park, K.; Park, Y. A Machine Learning Attack-Resistant PUF-based Robust and Efficient Mutual Authentication Scheme in Fog-enabled IoT Environments. *IEEE Internet Things J.* **2025**, *12*, 20652–20669. [CrossRef]
31. Yu, S.; Park, Y. A robust authentication protocol for wireless medical sensor networks using blockchain and physically unclonable functions. *IEEE Internet Things J.* **2022**, *9*, 20214–20228. [CrossRef]
32. Yu, S.; Das, A.K.; Park, Y.; Lorenz, P. SLAP-IoD: Secure and lightweight authentication protocol using physical unclonable functions for internet of drones in smart city environments. *IEEE Trans. Veh. Technol.* **2022**, *71*, 10374–10388. [CrossRef]
33. Chuang, K.H.; Bury, E.; Degraeve, R.; Kaczer, B.; Linten, D.; Verbauwhede, I. A physically unclonable function using soft oxide breakdown featuring 0% native BER and 51.8 fJ/bit in 40-nm CMOS. *IEEE J. Solid-State Circuits* **2019**, *54*, 2765–2776. [CrossRef]
34. Wang, W.C.; Yona, Y.; Diggavi, S.N.; Gupta, P. Design and analysis of stability-guaranteed PUFs. *IEEE Trans. Inf. Forensics Secur.* **2017**, *13*, 978–992. [CrossRef]
35. Alruwaili, O.; Alotaibi, F.M.; Tanveer, M.; Chaoui, S.; Armghan, A. PSAF-IoT: Physically secure authentication framework for the Internet of Things. *IEEE Access* **2024**, *12*, 78549–78561. [CrossRef]
36. Sarbishaei, G.; Modarres, A.M.A.; Jowshan, F.; Khakzad, F.Z.; Mokhtari, H. Smart home security: An efficient multi-factor authentication protocol. *IEEE Access* **2024**, *12*, 106253–106272. [CrossRef]
37. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **2003**, *29*, 198–208. [CrossRef]
38. Armando, A.; Basin, D.; Boichut, Y.; Chevalier, Y.; Compagna, L.; Cuéllar, J.; Drielsma, P.H.; Héam, P.C.; Kouchnarenko, O.; Mantovani, J.; et al. The AVISPA tool for the automated validation of internet security protocols and applications. In *Proceedings of the International Conference on Computer Aided Verification*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 281–285.
39. Abdalla, M.; Fouque, P.A.; Pointcheval, D. Password-based authenticated key exchange in the three-party setting. In *Proceedings of the International Workshop on Public Key Cryptography*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 65–84.
40. Ju, S.; Park, H.; Son, S.; Kim, H.; Park, Y.; Park, Y. Blockchain-assisted secure and lightweight authentication scheme for multi-server internet of drones environments. *Mathematics* **2024**, *12*, 3965. [CrossRef]
41. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
42. Ghosh, H.; Das, D.; Bagchi, S. UMAPE: A UAV-Assisted Secure Mutual Authentication Protocol Using Edge-Computing for Enhanced IoV Systems. In Proceedings of the 2025 IEEE 25th International Symposium on Cluster, Cloud and Internet Computing (CCGrid), Tromsø, Norway, 19–22 May 2025; pp. 205–214.
43. Zhang, J.; Cui, J.; Zhong, H.; Bolodurina, I.; Liu, L. Intelligent drone-assisted anonymous authentication and key agreement for 5G/B5G vehicular ad-hoc networks. *IEEE Trans. Netw. Sci. Eng.* **2020**, *8*, 2982–2994. [CrossRef]
44. Gupta, K.; Sharma, T. Changing trends in computer architecture: A comprehensive analysis of arm and x86 processors. *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.* **2021**, *7*, 619–631. [CrossRef]
45. Mejias, L.; Diguet, J.P.; Dezan, C.; Campbell, D.; Kok, J.; Coppin, G. Embedded computation architectures for autonomy in unmanned aircraft systems (UAS). *Sensors* **2021**, *21*, 1115. [CrossRef]
46. Ahmed, F.; Jenihhin, M. A survey on UAV computing platforms: A hardware reliability perspective. *Sensors* **2022**, *22*, 6286. [CrossRef]
47. Madhuvanthi, T.; Revathi, A. A survey on UAV network for secure communication and attack detection: A focus on Q-learning, blockchain, IRS and mmWave technologies. *KSII Trans. Internet Inf. Syst. (TIIS)* **2024**, *18*, 779–800.
48. Haidar, F.; Makassikis, M.; Sall, M.; Bakhti, H.; Kaiser, A.; Lonc, B. Experimentation and assessment of pseudonym certificate management and misbehavior detection in C-ITS. *IEEE Open J. Intell. Transp. Syst.* **2021**, *2*, 128–139. [CrossRef]
49. Xia, Y.; Qi, R.; Ji, S.; Shen, J.; Miao, T.; Wang, H. PUF-Assisted Lightweight Group Authentication and Key Agreement Protocol in Smart Home. *Wirel. Commun. Mob. Comput.* **2022**, *2022*, 8865158. [CrossRef]
50. Kenney, J.B. Dedicated short-range communications (DSRC) standards in the United States. *Proc. IEEE* **2011**, *99*, 1162–1182. [CrossRef]
51. Bazzi, A. Congestion control mechanisms in IEEE 802.11 p and sidelink C-V2X. In Proceedings of the 2019 53rd Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, 3–6 November 2019; pp. 1125–1130.
52. Sepulcre, M.; Gonzalez-Martin, M.; Gozalvez, J.; Molina-Masegosa, R.; Coll-Perales, B. Analytical models of the performance of IEEE 802.11 p vehicle to vehicle communications. *IEEE Trans. Veh. Technol.* **2021**, *71*, 713–724. [CrossRef]
53. Gozálvez, J.; Sepulcre, M.; Bauza, R. IEEE 802.11 p vehicle to infrastructure communications in urban environments. *IEEE Commun. Mag.* **2012**, *50*, 176–183. [CrossRef]